Document Title	Diagnostic Extract Template
Document Category	Template specification (TPS)
Document UID	673
Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R26-11

Document Change History

Date	Release	Changed by	Description
2026-11-26	R26-11	AUTOSAR Release Management	• Retire unused encodings • minor corrections / clarifications / editorial changes
2025-11-27	R25-11	AUTOSAR Release Management	• Improve support for post-build variance for OBD • Platform-independent support for extended data records • Add imposition time for AP • minor corrections / clarifications / editorial changes
2024-11-27	R24-11	AUTOSAR Release Management	• Improved handling of variants • Improved handling of security events • minor corrections / clarifications / editorial changes
2023-11-23	R23-11	AUTOSAR Release Management	• Improved diagnostic authentication • minor corrections / clarifications / editorial changes
2022-11-24	R22-11	AUTOSAR Release Management	• Support for nested data structures • Rework of ROE • Improvement of environmental condition • minor corrections / clarifications / editorial changes
2021-11-25	R21-11	AUTOSAR Release Management	• Support for Authentication service • Support for CpSoftwareCluster • minor corrections / clarifications / editorial changes



△

2020-11-30	R20-11	AUTOSAR Release Management	• Handling of Security Events • minor corrections / clarifications / editorial changes
2019-11-28	R19-11	AUTOSAR Release Management	• Increase support for variant handling • Improved Dem upstream mapping • Support for custom service instances • minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2017-12-08	4.3.1	AUTOSAR Release Management	• minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2016-11-30	4.3.0	AUTOSAR Release Management	• Support for OBD • Support for J1939 • Support for Fim configuration • Support for environmental conditions • Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2015-07-31	4.2.2	AUTOSAR Release Management	• Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2014-10-31	4.2.1	AUTOSAR Release Management	• Initial Release

Disclaimer

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

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

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

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Introduction 14
1.1 Overview . . . 14
1.1.1 OEM . . . 15
1.1.2 Application Developer . . . 16
1.1.3 ECU-Supplier . . . 16
1.1.4 Exchanging of Files . . . 17
1.1.5 Relationship to software-component Service Needs . . . 18
1.1.6 Recommendation and Hints . . . 19
1.1.7 Limitations . . . 19
1.2 Modeling of Diagnostic Configuration on classic and adaptive Platform . 19
1.3 Scope . . . 20
1.4 Document Conventions . . . 21
1.5 About Tables . . . 23
2 Use Cases 24
2.1 Use cases for diagnostic data exchange . . . 24
2.2 Configuration of DCM . . . 24
2.3 Configuration of DEM . . . 24
2.4 Configuration of the Fim . . . 26
2.4.1 Model Function Inhibition . . . 26
2.4.2 Model Fim configuration before Dem exists . . . 27
2.5 Configuration of J1939 Diagnostics . . . 27
2.5.1 Modeling of J1939 Diagnostic Aspects independent of the Deployment . . . 28
2.5.2 J1939 Diagnostic Content modeled in the Diagnostic Extract . . . 28
3 Conceptual Background 29
3.1 Definition of relevant Diagnostic Elements . . . 29
3.2 Abstraction from EcuC Level . . . 29
3.3 Independence of Definition . . . 30
3.3.1 Enabling separation of elements over several physical files . . . 30
3.3.2 Use of self-contained mapping elements . . . 30
4 Diagnostic Resources 31
4.1 Overview . . . 31
4.2 Common Meta Model Elements . . . 31
4.2.1 Overview . . . 31
4.2.2 Diagnostic Data Identifier . . . 32
4.2.3 Diagnostic Parameter . . . 34
4.2.3.1 Overview . . . 34
4.2.3.2 Size of a Diagnostic Parameter . . . 37
4.2.3.3 Start Position of a Diagnostic Parameter . . . 37
4.2.3.4 Content of a Diagnostic Parameter . . . 38

4.2.3.5 Diagnostic Parameter as Reference Target . . . 39
4.2.4 Diagnostic Data Element . . . 40
4.2.4.1 Usage of SwDataDefProps . . . 44
4.2.5 Definition of Arrays on the level of DiagnosticDataElement . . . 49
4.2.5.1 Definition of textual Strings . . . 50
4.2.6 Textual Documentation . . . 51
4.2.7 Diagnostic Contribution . . . 54
4.2.8 Diagnostic Protocol . . . 58
4.2.8.1 Implications on the Workflow to create a Diagnostic Extract . . . 61
4.2.9 Diagnostic Common Properties . . . 63
4.3 Diagnostic Services . . . 68
4.3.1 Introduction . . . 68
4.3.2 Diagnostic Service Instance vs. Diagnostic Service Class . . . 68
4.3.3 Diagnostic Access Permission . . . 72
4.3.4 Diagnostic Session . . . 74
4.3.5 Diagnostic Security Level . . . 76
4.3.6 Diagnostic Authentication . . . 76
4.3.7 Diagnostic Environmental Condition . . . 79
4.3.7.1 Environmental Condition Formula . . . 80
4.3.7.2 Atomic Conditions . . . 83
4.3.8 Diagnostic Services supported by AUTOSAR . . . 91
4.3.8.1 SessionControl . . . 93
4.3.8.2 SecurityAccess . . . 95
4.3.8.3 Authentication . . . 98
4.3.8.4 EcuReset . . . 102
4.3.8.5 CommunicationControl . . . 105
4.3.8.6 ControlDTCSetting . . . 112
4.3.8.7 DataByIdentifier . . . 113
4.3.8.8 IOControl . . . 118
4.3.8.9 RoutineControl . . . 122
4.3.8.10 DynamicallyDefineDataIdentifier . . . 128
4.3.8.11 ReadDataByPeriodicIdentifier . . . 131
4.3.8.12 ResponseOnEvent . . . 134
4.3.8.13 ReadDTCInformation . . . 137
4.3.8.14 ClearDiagnosticInformation . . . 139
4.3.8.15 Memory Services . . . 140
4.3.8.16 RequestFileTransfer . . . 148
4.3.9 OBD Diagnostic Services supported by AUTOSAR . . . 149
4.3.9.1 OBD Mode 0x01 (RequestCurrentPowertrainDiagnosticData) . . . 152
4.3.9.2 OBD Mode 0x02 (RequestPowertrainFreezeFrameData) . . . 154
4.3.9.3 OBD Mode 0x03 / 0x07 (RequestEmissionRelatedDiagnostic-TroubleCodes) . . . 156

4.3.9.4 OBD Mode 0x04 (ClearResetEmissionRelatedDiagnosticInformation) . . . 157
4.3.9.5 OBD Mode 0x06 (RequestOnBoardMonitoringTestResults) . . . 158
4.3.9.6 OBD Mode 0x08 (RequestControlOfOnBoardDevice) . . . 159
4.3.9.7 OBD Mode 0x09 (RequestVehicleInformation) . . . 161
4.3.9.8 OBD Mode 0x0A (RequestEmissionRelatedDiagnosticTroubleCodesPermanentStatus) . . . 163
4.3.10 UDS Diagnostic Services for supporting WWH-OBD . . . 164
4.4 Diagnostic Event Handling . . . 165
4.4.1 Introduction . . . 165
4.4.2 Diagnostic Event . . . 166
4.4.2.1 Overview . . . 166
4.4.2.2 Textual description . . . 170
4.4.2.3 Associated Event Identification . . . 173
4.4.3 Diagnostic Trouble Code . . . 174
4.4.4 Diagnostic Memory Destination . . . 187
4.4.5 Diagnostic Extended Data Record . . . 191
4.4.6 Diagnostic Freeze Frame . . . 195
4.4.7 Diagnostic Condition . . . 197
4.4.8 Diagnostic Debouncing . . . 199
4.4.9 Diagnostic Condition Group . . . 204
4.4.10 Diagnostic Operation Cycle . . . 205
4.4.11 Diagnostic Aging . . . 206
4.4.12 Diagnostic Indicator . . . 207
4.4.13 Diagnostic Test Result . . . 208
4.4.14 OBD-related aspects of Dem Configuration . . . 211
4.4.14.1 Dem Configuration for OBD-II . . . 216
4.4.14.2 Dem Configuration for WWH-OBD . . . 217
4.5 Functional Inhibition . . . 217
4.5.1 Introduction . . . 217
4.5.2 Alias Events . . . 218
4.5.3 Function Identifier . . . 219
4.6 Diagnostics on J1939 . . . 222
4.6.1 Introduction . . . 222
4.6.2 Suspect Parameter Number . . . 223
4.6.3 Dem-related Modeling . . . 223
5 Diagnostic Mapping 227
5.1 Introduction . . . 227
5.2 Mapping of Diagnostic Services . . . 228
5.2.1 Diagnostic Service Data Mapping . . . 231
5.2.2 Diagnostic Service Data Mapping for OBD . . . 236
5.2.3 Diagnostic Service Software Mapping . . . 239
5.2.4 Diagnostic Secure Coding . . . 244

5.2.5 Diagnostic Authentication Transmit Certificate Mapping 245
5.3 Event-related Diagnostic Mapping 246
5.3.1 DiagnosticEvent to DtcUds Mapping 247
5.3.2 DiagnosticEvent to DiagnosticOperationCycle Mapping 248
5.3.3 DiagnosticEvent to DebounceAlgorithm Mapping 249
5.3.4 DiagnosticEvent to EnableConditionGroup Mapping 250
5.3.5 DiagnosticEvent to StorageConditionGroup Mapping 251
5.3.6 DiagnosticEvent to Port Mapping 252
5.3.7 DiagnosticOperationCycle to Port Mapping 254
5.3.8 DiagnosticEnableCondition to Port Mapping 256
5.3.9 DiagnosticStorageCondition to Port Mapping 257
5.3.10 Extended Data Record Mapping 258
5.3.10.1 Extended Data Record Data Provider Mapping 259
5.3.10.2 Extended Data Record Sender Port Mapping 262
5.3.10.3 Extended Data Record ServerPort Mapping 263
5.3.11 Master to Slave Event Mapping 264
5.3.12 Diagnostic Event to Security Event Mapping 265
5.3.13 Diagnostic Event to ISignalPort Mapping 269
5.4 Mapping of Function Inhibition 271
5.4.1 Mapping between Inhibition Source and Diagnostic Event 271
5.4.2 Alias Event Mapping 272
5.4.3 Mapping of Function Identifiers to Interested Software Components 275
5.4.4 Diagnostic IUMPR to Function Identifier Mapping 277
5.5 J1939-related DiagnosticMapping 278
5.5.1 SPN Mapping 278
5.5.2 Mapping between Software Components and Controller Applications 280
5.5.3 Mapping between DiagnosticEvent and J1939 DTC 281
5.6 Diagnostic Configuration of CP Software Clusters 282
5.6.1 Association of Diagnostic Mapping with a CP Software Cluster . . . 282
5.6.2 Mapping between DEXT elements and CP Software Cluster Resources 283
5.6.3 Mapping between Diagnostic Events and CP Software Cluster Resources 284
5.6.4 Mapping between Diagnostic Data Elements and CP Software Cluster Resources 285
5.6.5 Mapping between Diagnostic Routines and CP Software Cluster Resources 285
5.6.6 Mapping between Diagnostic Function Identifiers and CP Software Cluster Resources 286
A Reference Material 288
A.1 Imposition Times of Constraints 288
A.2 Abbreviations 288

B Examples 291
B.1 Examples of the Definition of nested Diagnostic Parameters 291
B.1.1 Simple Nesting Example 291
B.1.2 Simple Nesting Example with Gaps 292
B.1.3 Two-Level Nesting Example 293
B.1.4 Array of Struct Example 295
B.2 Examples of Diagnostic Service Data Mapping 297
B.2.1 Simple Structure 297
B.2.2 Nested Structure 298
B.2.3 Array of Structure 299
B.3 Examples of the Definition of Mature and De-Mature Condition 301
B.4 Examples of the Definition of semi-formal Textual Description 302
B.5 Examples of the Definition of Pre- and Run-Condition 302
C Mentioned Class Tables 304
D Glossary 348
E Change history of AUTOSAR traceable items 352
E.1 Traceable item history of this document according to AUTOSAR Release R4.2.1 352
E.1.1 Added Specification Items in R4.2.1 352
E.1.2 Added Constraints in R4.2.1 356
E.2 Traceable item history of this document according to AUTOSAR Release R4.2.2 357
E.2.1 Added Traceables in R4.2.2 357
E.2.2 Changed Traceables in R4.2.2 357
E.2.3 Deleted Traceables in R4.2.2 357
E.2.4 Added Constraints in R4.2.2 358
E.2.5 Changed Constraints in R4.2.2 358
E.2.6 Deleted Constraints in R4.2.2 358
E.3 Traceable item history of this document according to AUTOSAR Release R4.3.0 358
E.3.1 Added Traceables in R4.3.0 358
E.3.2 Changed Traceables in R4.3.0 360
E.3.3 Deleted Traceables in R4.3.0 360
E.3.4 Added Constraints in R4.3.0 360
E.3.5 Changed Constraints in R4.3.0 362
E.3.6 Deleted Constraints in R4.3.0 362
E.4 Traceable item history of this document according to AUTOSAR Release R4.3.1 362
E.4.1 Added Specification Items in 4.3.1 362
E.4.2 Changed Specification Items in 4.3.1 362
E.4.3 Deleted Specification Items in 4.3.1 362
E.4.4 Added Constraints in 4.3.1 363

E.4.5 Changed Constraints in 4.3.1 . . . 363
E.4.6 Deleted Constraints in 4.3.1 . . . 363
E.5 Traceable item history of this document according to AUTOSAR Release R4.4.0 . . . 364
E.5.1 Added Specification Items in 4.4.0 . . . 364
E.5.2 Changed Specification Items in 4.4.0 . . . 364
E.5.3 Deleted Specification Items in 4.4.0 . . . 364
E.5.4 Added Constraints in 4.4.0 . . . 365
E.5.5 Changed Constraints in 4.4.0 . . . 365
E.5.6 Deleted Constraints in 4.4.0 . . . 365
E.6 Traceable item history of this document according to AUTOSAR Release R19-11 . . . 366
E.6.1 Added Specification Items in 19-11 . . . 366
E.6.2 Changed Specification Items in 19-11 . . . 366
E.6.3 Deleted Specification Items in 19-11 . . . 366
E.6.4 Added Constraints in 19-11 . . . 367
E.6.5 Changed Constraints in 19-11 . . . 367
E.6.6 Deleted Constraints in 19-11 . . . 367
E.7 Traceable item history of this document according to AUTOSAR Release R20-11 . . . 367
E.7.1 Added Specification Items in R20-11 . . . 367
E.7.2 Changed Specification Items in R20-11 . . . 368
E.7.3 Deleted Specification Items in R20-11 . . . 368
E.7.4 Added Constraints in R20-11 . . . 368
E.7.5 Changed Constraints in R20-11 . . . 371
E.7.6 Deleted Constraints in R20-11 . . . 371
E.8 Traceable item history of this document according to AUTOSAR Release R21-11 . . . 372
E.8.1 Added Specification Items in R21-11 . . . 372
E.8.2 Changed Specification Items in R21-11 . . . 372
E.8.3 Deleted Specification Items in R21-11 . . . 373
E.8.4 Added Constraints in R21-11 . . . 373
E.8.5 Changed Constraints in R21-11 . . . 373
E.8.6 Deleted Constraints in R21-11 . . . 374
E.9 Traceable item history of this document according to AUTOSAR Release R22-11 . . . 374
E.9.1 Added Specification Items in R22-11 . . . 374
E.9.2 Changed Specification Items in R22-11 . . . 375
E.9.3 Deleted Specification Items in R22-11 . . . 376
E.9.4 Added Constraints in R22-11 . . . 376
E.9.5 Changed Constraints in R22-11 . . . 376
E.9.6 Deleted Constraints in R22-11 . . . 377
E.10 Traceable item history of this document according to AUTOSAR Release R23-11 . . . 377

E.10.1 Added Specification Items in R23-11 . . . 377
E.10.2 Changed Specification Items in R23-11 . . . 378
E.10.3 Deleted Specification Items in R23-11 . . . 378
E.10.4 Added Constraints in R23-11 . . . 379
E.10.5 Changed Constraints in R23-11 . . . 379
E.10.6 Deleted Constraints in R23-11 . . . 379
E.10.7 Added Advisories in R23-11 . . . 380
E.10.8 Changed Advisories in R23-11 . . . 380
E.10.9 Deleted Advisories in R23-11 . . . 380
E.11 Traceable item history of this document according to AUTOSAR Release R24-11 . . . 380
E.11.1 Added Specification Items in R24-11 . . . 380
E.11.2 Changed Specification Items in R24-11 . . . 380
E.11.3 Deleted Specification Items in R24-11 . . . 380
E.11.4 Added Constraints in R24-11 . . . 381
E.11.5 Changed Constraints in R24-11 . . . 381
E.11.6 Deleted Constraints in R24-11 . . . 381
E.11.7 Added Advisories in R24-11 . . . 381
E.11.8 Changed Advisories in R24-11 . . . 381
E.11.9 Deleted Advisories in R24-11 . . . 382
E.12 Traceable item history of this document according to AUTOSAR Release R25-11 . . . 382
E.12.1 Added Specification Items in R25-11 . . . 382
E.12.2 Changed Specification Items in R25-11 . . . 382
E.12.3 Deleted Specification Items in R25-11 . . . 382
E.12.4 Added Constraints in R25-11 . . . 383
E.12.5 Changed Constraints in R25-11 . . . 383
E.12.6 Deleted Constraints in R25-11 . . . 384
E.12.7 Added Advisories in R25-11 . . . 384
E.12.8 Changed Advisories in R25-11 . . . 384
E.12.9 Deleted Advisories in R25-11 . . . 384
E.13 Traceable item history of this document according to AUTOSAR Release R26-11 . . . 385
E.13.1 Added Specification Items in R26-11 . . . 385
E.13.2 Changed Specification Items in R26-11 . . . 385
E.13.3 Deleted Specification Items in R26-11 . . . 385
E.13.4 Added Constraints in R26-11 . . . 386
E.13.5 Changed Constraints in R26-11 . . . 386
E.13.6 Deleted Constraints in R26-11 . . . 386
E.13.7 Added Advisories in R26-11 . . . 386
E.13.8 Changed Advisories in R26-11 . . . 386
E.13.9 Deleted Advisories in R26-11 . . . 386

F Modeling of InstanceRef 387
F.1 Introduction . . . 387
F.2 Modeling . . . 388
G Upstream Mapping 391
G.1 Introduction . . . 391
G.2 Dcm . . . 392
G.3 Dem . . . 667
G.4 FiM . . . 768
G.5 IdsM . . . 772
G.6 J1939Dcm . . . 772
G.7 KeyM . . . 781
H Non-Splitable Elements in the Scope of this Document 783
I Variation Points in the Scope of this Document 784

References

[1] Specification of RTE Software
AUTOSAR_CP_SWS_RTE

[2] Layered Software Architecture
AUTOSAR_CP_EXP_LayeredSoftwareArchitecture

[3] Specification of ECU Configuration
AUTOSAR_CP_TPS_ECUConfiguration

[4] ASAM MCD 2D ODX
http://www.asam.net
ASAM MCD-2D ODX v2.2.0.pdf

[5] System Template
AUTOSAR_CP_TPS_SystemTemplate

[6] Specification of ECU Configuration Parameters (XML)
AUTOSAR_CP_MOD_ECUConfigurationParameters

[7] Specification of Manifest
AUTOSAR_AP_TPS_ManifestSpecification

[8] Software Component Template
AUTOSAR_CP_TPS_SoftwareComponentTemplate

[9] Basic Software Module Description Template
AUTOSAR_CP_TPS_BSWModuleDescriptionTemplate

[10] Specification of Diagnostic Communication Manager
AUTOSAR_CP_SWS_DiagnosticCommunicationManager

[11] Specification of Diagnostic Event Manager
AUTOSAR_CP_SWS_DiagnosticEventManager

[12] Standardization Template
AUTOSAR_FO_TPS_StandardizationTemplate

[13] Specification of Function Inhibition Manager
AUTOSAR_CP_SWS_FunctionInhibitionManager

[14] SAE J1939 – Serial Control and Communications Heavy Duty Vehicle Network

[15] Generic Structure Template
AUTOSAR_FO_TPS_GenericStructureTemplate

[16] Collection of blueprints for AUTOSAR M1 models
AUTOSAR_FO_MOD_GeneralBlueprints

[17] ISO 14229-1 – Unified diagnostic services (UDS) – Part 1: Specification and requirements (Release 2006-12)
https://www.iso.org

[18] ISO 26021-2 – Road vehicles – End-of-life activation of on-board pyrotechnic devices – Part 2: Communication requirements
https://www.iso.org

[19] ISO 15031-5 – Road vehicles – Communication between vehicle and external equipment for emission-related diagnostic – Part 5: Emission-related diagnostic services.
https://www.iso.org/standard/66368.html

[20] ISO 27145-3:2012 – Road vehicles – Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements – Part 3: Common message dictionary
https://www.iso.org/standard/46277.html

[21] SAE J1939-73 Application Layer – Diagnostics

[22] SAE J1979 – E/E Diagnostic Test Modes

[23] Security Extract Template
AUTOSAR_FO_TPS_SecurityExtractTemplate

[24] Software Process Engineering Meta-Model Specification
http://www.omg.org/spec/SPEM/2.0/

1 Introduction

for R26-11

Note: The life cycle of the stereotype ≪atpSplitable≫ will end with the AUTOSAR Release R26-11. Any remaining references to ≪atpSplitable≫ in this document will be removed in release R27-05.

1.1 Overview

The distributed nature of an AUTOSAR ECU development requires an optimized capturing of information. In particular, diagnostic information (i.e. DEM and DCM configuration) shall be captured only once by the person with the best knowledge and therefore being able to take responsibility better than one centralized individual.

In the configuration approach before the advent of the DiagnosticExtract, the Basic Software Modules DCM and DEM are entirely configured centrally.

During integration, some ApplicationSwComponentTypes above the RTE [1, SWS RTE] (Application Software) introduce PortPrototypes to be connected to the ServiceSwComponentTypes that represent the service-related basic software modules, see [2, EXP Layered Software Architecture].

Additionally, ApplicationSwComponentTypes express needs which shall be fulfilled by the ServiceSwComponentTypes.

The market shows a high demand for transferring diagnostic requirements of the OEM-specific configuration process to their tier-1 suppliers.

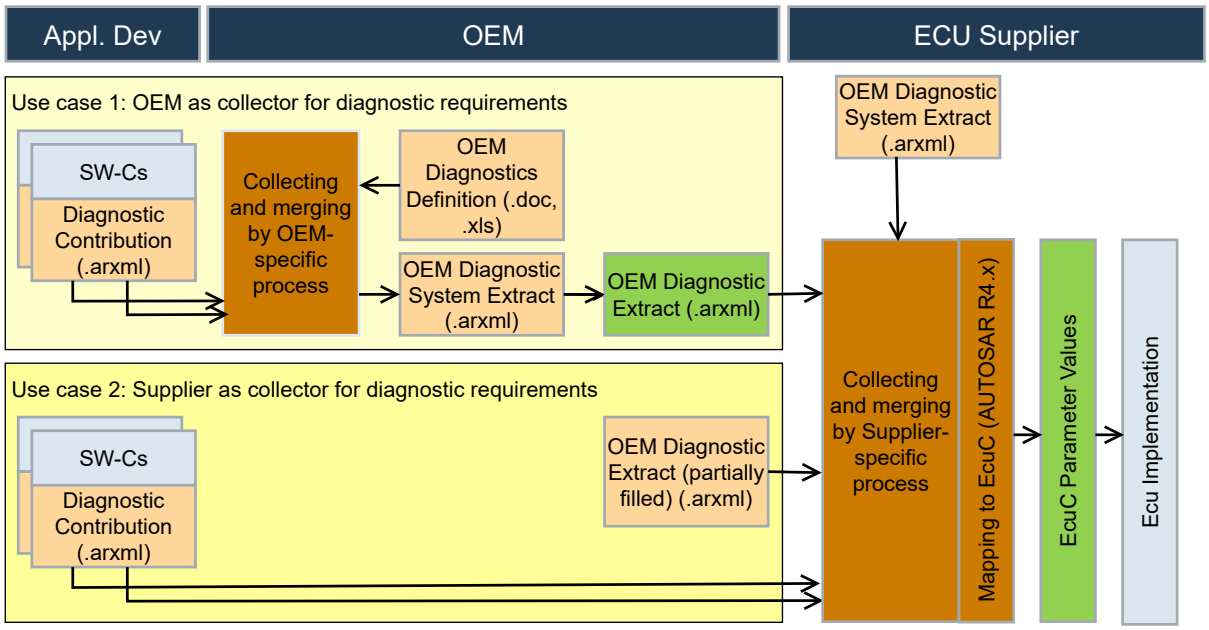

Figure 1.1: Scope of this document in the ECU Development work-flow

In the past, due to the absence of integral options, many file formats like ODX or EcuC (see [3, TPS Ecuc Configuration]) are often used. But neither ODX nor EcuC is well suited to transfer this information.

For example, [4, ODX] lacks in fault memory details and EcuC (which was never designed for becoming the vehicle for data exchange between different organizations) has a very generic nature that renders the enforcement of a strict model formalization very difficult.

On top of that, the integration of EcuC definitions into an existing configuration (especially the PDUs) cannot be fully automated.

Therefore, the obvious solution approach has been to define a dedicated standardized AUTOSAR exchange format on diagnostic functionality that can be used similar to a System Description, formalized as an ARXML file.

In this spirit, the configuration of diagnostic functionality becomes similar to the configuration of the communication part within the system description, see [5, TPS System Template].

Figure 1.1 shows the configuration process of diagnostics for two generalized use cases. This process involves three parties:

- **OEM** or diagnostic requester
- **Application Developer** or Application Developer
- **ECU-Supplier** or integrator

The specific role of these contributors to the diagnostic extract is in detail explained in the following sub-chapters.

1.1.1 OEM

The **OEM** or requester of diagnostic data uses the DiagnosticExtract to define the diagnostic interfaces of one or multiple ECUs. It may also define some InternalBehaviors as requirements for the **ECU-Supplier** or **Application Developer**

- Defines the values of the DTCs
- Defines the UDS services and sub-services supported by the ECUs
- Defines the required events needed by a specific composition implemented by an **Application Developer**

NOTE: This list represents an example; this document does not define a specific ownership of each element.

In the first use case, the DiagnosticExtract is used to exchange information which is transformed into the EcuC (M2 to M1 mapping, see also [3, TPS Ecu Configuration] and [6, MOD Ecu Configuration Parameters]).

Second, the **OEM** uses the DiagnosticExtract to document requirements to be implemented by a supplier. These requirements are expressed in textual language and can not be mapped directly to any EcuC parameters (no M2 to M1 mapping possible).

1.1.2 Application Developer

The Application Developers implement their **software-components** with the corresponding **software-component description**. The role "Application Developer" can be assumed by both an OEM and a supplier. In other words, both OEM and supplier may contribute application software to a given ECU.

With the introduction of this concept, the Application Developer has the possibility to provide diagnostic information relevant to the software-components as part of the DiagnosticExtract.

The Application Developer may also receive some input as requirement from the **OEM** within the DiagnosticExtract in textual form as for example:

Example 1.1

Definition of the content of a specific ReadDataByIdentifier implemented by this software-component

Example 1.2

Definition of the events needed for this software-component

NOTE: Only mentioned as as example content. This document does not define a specific ownership of each element.

In the first use case, the Application Developer defines the parameters of a specific ReadDataByIdentifier, i.e. the content of the diagnostic request but not the DID itself. The DID of this command will usually be defined by the **OEM**.

Secondly, the software-component events including information like Debouncing and OperationCycle may be defined by the **Application Developer**. The **Application Developer** may also define events and diagnostic jobs which are not needed by a specific **OEM** but for another one.

Suppliers may use the same software for multiple **OEM**s and need to reuse it. This implies that some DiagnosticExtract information coming from a software-component may be ignored during the integration if not needed for a specific project.

1.1.3 ECU-Supplier

The **ECU-Supplier** or integrator receives one or several DiagnosticExtract files from the **OEM** and from multiple **Application Developers**. The main goal of the **Integrator** is to integrate all delivered DiagnosticExtract and to generate the EcuC from it.

Since this concept does not define a specific ownership for each element like DIDs, parameters of a UDS service, Events, Sessions, etc. the integrator has to ensure that the complete information is still valid after merging it.

- Mapping of DTCs to Events
- Merge of Events
- Mapping of services

Some DTCs may already be mapped to events - especially in cases where both come from the same party. But if the DTCs are defined by the **OEM** and the software implemented by other supplier acting as an **Application Developer** the integrator has to ensure that both are mapped together.

In some cases, an Event may be defined multiple times. An **OEM** defines the Events which shall be implemented by an **Application Developer**. A Supplier implements a software-component which will be used in multiple projects and which also detects this type of error and also defines this same event.

Both events may have different naming but the same meaning. The integrator has to detect this redundancy during the integration and merge them together.

In another case, the **OEM** requires a specific `ReadDataByIdentifier` and an **Application Developer** implements it. If the implementation is performed for one specific project only, the **Application Developer** may map the DID from the **OEM** to the already defined job in their software-component.

In other cases, in which the **Application Developer** implements a generic diagnostic job, it will be a task of the **ECU-Supplier** to merge this information and to map the jobs to the corresponding DID.

1.1.4 Exchanging of Files

During an ECU development project, the three main roles (OEM, Application Developer, ECU Integrator) exchange `DiagnosticExtract` files.

The timing and frequency of exchanges and the content in each of these exchanged files is highly dependent on the individual project setup and situation.

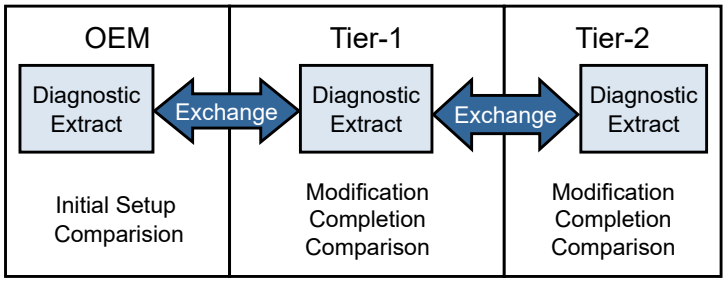

Figure 1.2: Exchange of diagnostic configuration between OEM, Tier-1, and Tier-2

Therefore, the DiagnosticExtract format has been designed to allow for gradual enrichment of definitions contributed at largely arbitrary points in time by the different roles in order to meet the needs of “Decentralized Configuration”.

For any exchange path between any two roles, the same file format based on the DiagnosticExtract template is used. It is then up to a company specific process and tooling to merge the collected DiagnosticExtract files while resolving conflicts (contradictions, redundancies etc.).

As final result, a consistent and complete DiagnosticExtract file is available which is input to derivation of the configuration for the diagnostic modules of the basic software.

Even after the DiagnosticExtract has been fully integrated and is ready to go for deriving the configuration of the diagnostic stack on EcuC, level it is still foreseen to feed it back to e.g. an OEM.

In this case the OEM has the ability to review the configuration of the diagnostic stack on the level of the diagnostic extract.

At some point, this information may also be taken to (directly or via indirection of other formats) create a configuration of a diagnostic client.

1.1.5 Relationship to software-component Service Needs

Before the introduction of the Diagnostic Extract, Service Needs were used to describe diagnostic requirements on software-component level. These configuration requirements are referenced to the related basic software module DCM or OEM in order to provide the corresponding configuration on basic software level.

The usage of ServiceNeeds is only possible on AtomicSwComponentType level whereas the assignment of diagnostic demands needs to be possible on the level of CompositionSwComponentTypes.

The software-component Service Needs within the software-component Description are still to be used along with the DiagnosticExtract in order to annotate the software-component ports which are relevant for further mapping and handling as defined by the DiagnosticExtract.

From software-component developer’s perspective, the DiagnosticExtract is therefore used partially as substitution and partially as extension of the software-component Service Needs. The reasoning for “substitution” is the avoidance of redundant diagnostic definitions.

Since some diagnostic properties potentially definable by software-component Developers are not covered in software-component Service Needs, the DiagnosticExtract can also be viewed as “extension” to the original purpose of software-component Service Needs.

1.1.6 Recommendation and Hints

Multiple parties may have different understanding of which parts shall be provided by each one. There is no defined rule to indicate who is responsible for each part. At the end, it is the **ECU-Supplier** in his role as integrator who has to ensure that all mappings are done and that the ECU runs as expected by the **OEM**.

If the **OEM** does not have own diagnostic requirements, the **ECU-Supplier** has to provide the complete DiagnosticExtract. In this case, the **OEM** may only receive the DiagnosticExtract as part of the delivery. The process itself how the parties work with this format is not defined within this specification.

Figure 1.1 shows a recommended way how to handle the DiagnosticExtract between the different parties. In use case 1, some software-components are implemented by the **OEM** (or by a Supplier of the OEM) and the first merging of DiagnosticExtract data occurs at the **OEM**.

In use case 2, the **OEM** provides the diagnostic requirements via DiagnosticExtract and multiple **Application Developer**s provide information related to their implementation, the merging is performed completely by the **ECU-Supplier**.

Also, a combination of use cases 1 and 2 is allowed. Also, the **ECU-Supplier** may implement some part of the software including their corresponding DiagnosticExtract.

1.1.7 Limitations

This release of the DiagnosticExtract template focuses on defining diagnostic requirements a **single ECU** only. For the time being, distributed diagnostic functionality for a system or partial system consisting of multiple ECUs cannot be defined using the DiagnosticExtract template.

In future releases, the DiagnosticExtract template may be extended to also cover configuration of distributed diagnostic functionality. Similar to the description of communication dependencies in the System Description, it may then be possible to describe diagnostic demands on system level to derive the diagnostic demands for a specific ECU from this description.

1.2 Modeling of Diagnostic Configuration on classic and adaptive Platform

The configuration of the “external behavior” of the AUTOSAR diagnostic stack is done by means of the so-called DiagnosticExtract. The DiagnosticExtract in general extends to both the *AUTOSAR classic platform* and the *AUTOSAR adaptive platform*.

In particular, some parts of the DiagnosticExtract apply only for the *AUTOSAR classic platform*, some parts are shared between the *AUTOSAR classic platform* and the *AUTOSAR adaptive platform*, and some parts apply only for the *AUTOSAR adaptive platform*.

The parts that are specific to the *AUTOSAR classic platform*, and the parts that apply to both the *AUTOSAR classic platform* and the *AUTOSAR adaptive platform* are documented in the deliverable "TPS Diagnostic Extract Template", i.e. this document.

For technical reasons, the parts that apply to the *AUTOSAR adaptive platform* cannot be included in this document, "TPS Diagnostic Extract Template" and are consequently described in the document [7, TPS Manifest Specification].

The parts about the configuration of Diagnostic Management, as described in "TPS Manifest Specification", fit seamlessly into the same modeling framework that is used on the *AUTOSAR classic platform*. This means that the specific parts for the *AUTOSAR adaptive platform* utilize the shared parts that are described in the "TPS Diagnostic Extract Template".

In other words, readers who want to focus on DiagnosticExtract specifically for application on the *AUTOSAR classic platform* may concentrate on reading the "TPS Diagnostic Extract Template".

On the other hand, readers who'd like to understand how the DiagnosticExtract works on the *AUTOSAR adaptive platform* will have to read the diagnostics-related chapters of the "TPS Manifest Specification", plus the relevant parts of the "TPS Diagnostic Extract Template".

In this document, the applicability to the different platforms is formally specified by the definition of constraints. In particular, constraints announce a relevance for the *AUTOSAR adaptive platform* by defining a respective imposition time for the *AUTOSAR adaptive platform*. As a random example, please refer to [constr_1792].

1.3 Scope

This document describes the formal description of contributions to the diagnostic configuration.

On the level of meta-modeling, the content described in this document conceptually relates to the definition of SwcServiceDependency or BswServiceDependency, as defined by the [8, TPS Software Component Template] or [9, BSW Module Description Template].

Further relations exist to the specification of communication in AUTOSAR systems as described by the [5, TPS System Template].

Further relations exist to the specification of the [10, SWS Diagnostic Communication Manager] as well as to the [11, SWS Diagnostic Event Manager].

Please note that the configuration of service-oriented vehicle diagnostics (SOVD) is only supported on the *AUTOSAR adaptive platform*. For more details, please refer to [7, TPS Manifest Specification] Chapter 5.

The relation of the `DiagnosticExtract` to the rest of the AUTOSAR meta-model is sketched in Figure 1.3.

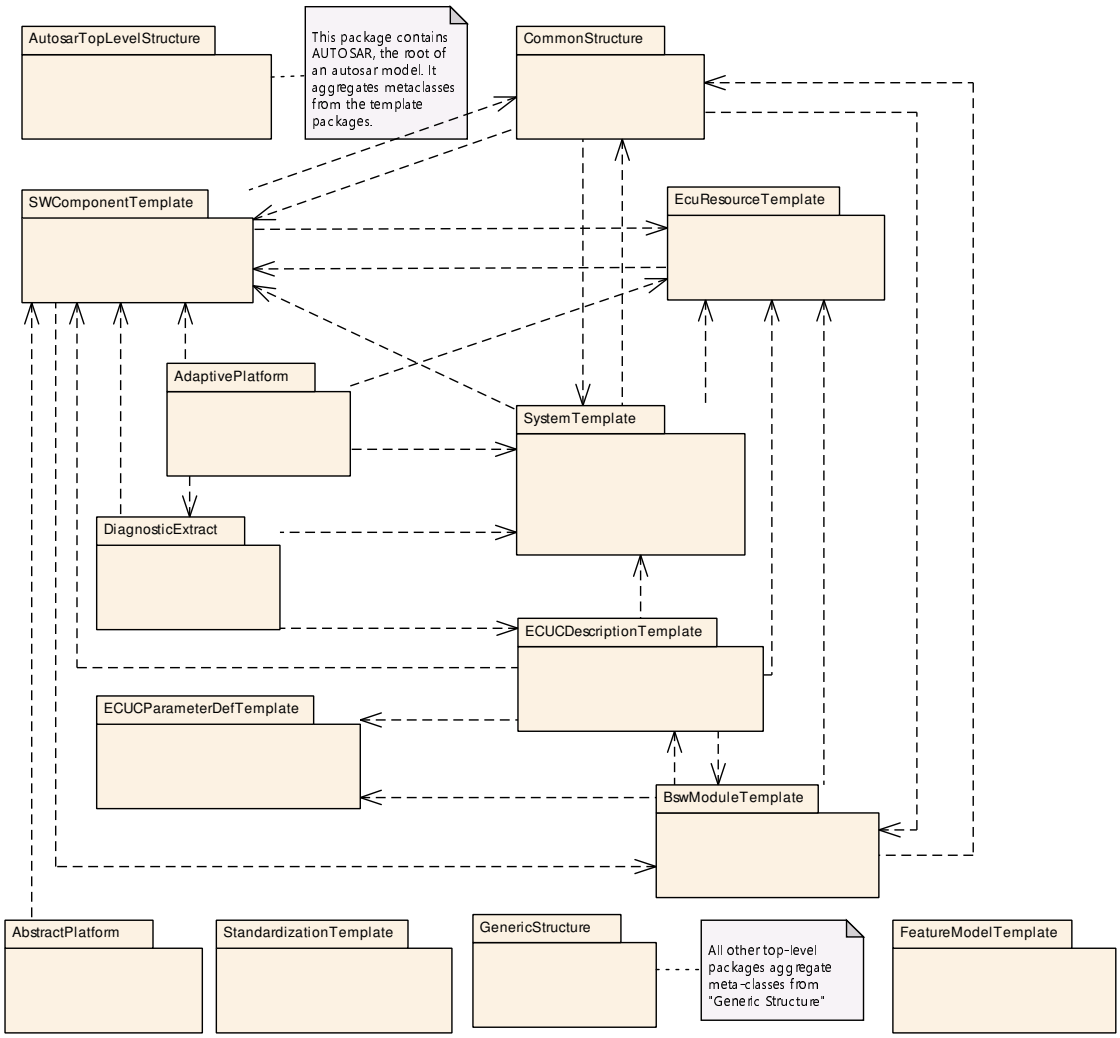

Figure 1.3: The relation of the `DiagnosticExtract` to the rest of the AUTOSAR meta-model

1.4 Document Conventions

Technical terms are typeset in mono spaced font, e.g. `PortPrototype`. As a general rule, plural forms of technical terms are created by adding “s” to the singular form, e.g. `PortPrototype`s. By this means the document resembles terminology used in the AUTOSAR XML Schema.

This document contains constraints in textual form that are distinguished from the rest of the text by a unique numerical constraint ID, a headline, and the actual constraint text starting after the ⌈ character and terminated by the ⌋ character.

The purpose of these constraints is to literally constrain the interpretation of the AUTOSAR meta-model such that it is possible to detect violations of the standardized behavior implemented in an instance of the meta-model (i.e. on M1 level).

Makers of AUTOSAR tools are encouraged to add the numerical ID of a constraint that corresponds to an M1 modeling issue as part of the diagnostic message issued by the tool.

The attributes of the classes introduced in this document are listed in form of class tables. They have the form shown in the example of the top-level element AUTOSAR:

Please note that constraints are not supposed to be enforceable at any given time in an AUTOSAR workflow. During the development of a model, constraints may legitimately be violated because an incomplete model will obviously show inconsistencies.

However, at specific points in the workflow, constraints shall be enforced as a safeguard against misconfiguration.

The points in the workflow where constraints shall be enforced, sometimes also known as the “imposition time” of the constraint, are different for each model category, e.g. on the classic platform, the constraints defined for software-components are typically enforced prior to the generation of the RTE while the constraints against the definition of an Ecu extract shall be applied when the Ecu configuration for the Com stack is created.

For each document, possible imposition times of constraints are defined and the imposition times are typically mentioned in the constraint themselves to give a proper orientation for implementers of AUTOSAR authoring tools.

Let AUTOSAR be an example of a typical class table. The first rows in the table have the following meaning:

Class: The name of the class as defined in the UML model.

Package: The UML package the class is defined in. This is only listed to help locating the class in the overall meta model.

Note: The comment the modeler gave for the class (class note). Stereotypes and UML tags of the class are also denoted here.

Base Classes: If applicable, the list of direct base classes.

The headers in the table have the following meaning:

Attribute: The name of an attribute of the class. Note that AUTOSAR does not distinguish between class attributes and owned association ends.

Type: The type of an attribute of the class.

Mul.: The assigned multiplicity of the attribute, i.e. how many instances of the given data type are associated with the attribute.

Kind: Specifies, whether the attribute is aggregated in the class (`aggr` aggregation), an UML attribute in the class (`attr` primitive attribute), or just referenced by it (`ref` reference). Instance references are also indicated (`iref` instance reference) in this field.

Note: The comment the modeler gave for the class attribute (role note). Stereotypes and UML tags of the class are also denoted here.

Please note that the chapters that start with a letter instead of a numerical value represent the appendix of the document. The purpose of the appendix is to support the explanation of certain aspects of the document and does not represent binding conventions of the standard.

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see [12, Standardization Template].

Please note: By intent, TPS documents (and their traceable items) do not trace up to an RS (requirement item).

1.5 About Tables

There is a number of tables in this document that describe the applicability or multiplicity of model elements. These tables contain empty cells, which represent a “not applicable” meaning. Model elements that correspond to an empty cell shall not exist in the context described by the respective table cell.

If the value of a table cell shall indicate that that either the existence or multiplicity of the represented model element shall be ignored anyway, the table cell gets the content “d/c” for “don’t care”. This means that if the respective model element exists, it shall be ignored in the downstream processing of the model.

2 Use Cases

2.1 Use cases for diagnostic data exchange

The primary purpose of the DiagnosticExtract is to exchange diagnostic data among the parties involved in diagnostic development. This exchange supports configuration of the DCM and DEM and provides descriptions of the application interfaces needed to implement diagnostic services and fault handling.

2.2 Configuration of DCM

As depicted in Figure 2.1, the configuration of the DCM includes the setup of diagnostic services and the assignment of data objects provided by one or more software components (e.g. Composition 1, Composition 2).

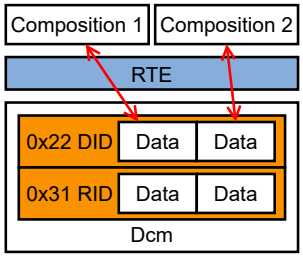

Figure 2.1: Assignment of DCM-related data objects

2.3 Configuration of DEM

As sketched in Figure 2.2, the configuration of the DEM includes fault memory data (DTCs and environmental data) and the assignment of corresponding data to one or more SwComponentTypes (e.g. Composition 1, Composition 2).

As already explained, the diagnostic development process is distributed among different parties. On the one hand, the OEM needs to describe the general requirements for a diagnostic system that have to be implemented by an ECU:

1. Depending on the diagnostic system, the OEM can provide a completely or partly-filled DiagnosticExtract including the description of PortInterfaces:
   - Integrator/SWC developer (OEM or Tier 1) is responsible for the completion (detailing of predefined diagnostic content).
   - Integrator/SWC developer (OEM or Tier 1) is responsible for the specific configuration of diagnostic content defined by himself.
   - New integration of updated Diagnostic descriptions by integrator.
2. Return the completed or partially completed DiagnosticExtract to OEM for:

- Documentation
- ECU testing
- Integration reviews
- Failure correction

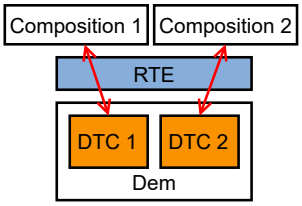

Figure 2.2: Assignment of DEM-related data objects

Use case examples:

- Configuration of UDS service 0x22 (ReadDataByIdentifier)
- Configuration of UDS service 0x2E (WriteDataByIdentifier)
- Configuration of UDS service 0x31 (RoutineControl)
- Configuration of UDS service 0x2F (I/O-Control)
- Configuration of DEM DTCs
- Configuration of Combined Events
- Mapping of events to DTCs
- Configuration of DTC-related environmental data
- Mapping of DEM Events to their corresponding Enable Conditions and Storage Conditions
- Configuration of general DCM parameters
- Description of diagnostic demands that are not relevant for code generation but have to be exchanged between OEM and Tier1 (e.g. set and reset condition for a DTC)

The OEM already provides a System Template for an ECU which describes the ECU Supplier SW parts as a CompositionSwComponentType where only inputs and outputs are known.

1. The OEM creates a DiagnosticExtract which describes the diagnostic interfaces of an ECU. Supported services are described (e.g. Read Data By Identifier/Write Data by Identifier/Routine Control) as well as their corresponding input/output parameters and return values (those that are optional in UDS standard).
2. Software-component developer at OEM or OEM software supplier develops software-components and also describes the diagnostic information using ServiceNeeds as diagnostic contribution.

3. The OEM Diagnostic Responsible for a project creates the mappings between the DiagnosticExtract and the SWC available on the OEM side (from 2.).
4. The OEM Diagnostic Responsible for a project creates the mappings between the DiagnosticExtract and the CompositionSwComponentTypes which will be implemented by the ECU Supplier or SW Developer.
5. The ECU Suppliers receive the ECU Extract including DiagnosticExtract from the OEM and imports it into the project.
6. As in step 2, the SWC Developer on the supplier or Tier 2 side describes the Diagnostic information using ServiceNeeds as diagnostic contribution.

The refinement of use cases is also depicted in Figure 2.3.

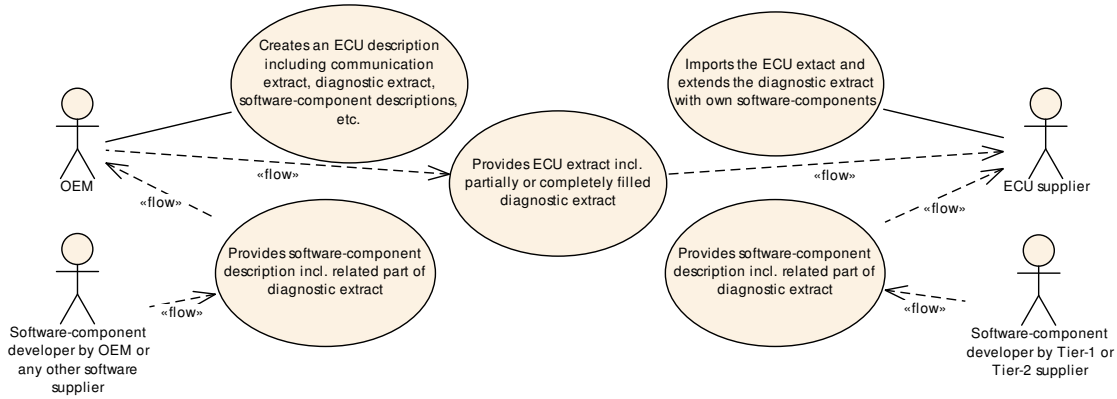

Figure 2.3: Workflow scenario for diagnostic data exchange

In the same way as in point 3: The ECU Supplier Diagnostic Responsible creates the mapping between the PortInterfaces of the DiagnosticExtract (from 5.) and the SwComponentTypes as provided in 6.

When indicators are used, those defined on BSW level in DEM may not be mapped automatically to their implementation on SWC level. This would then require a manual mapping step by the integrator to resolve the mismatch.

2.4 Configuration of the Fim

2.4.1 Model Function Inhibition

A typical use case for the FIM is the definition of “control mechanisms for software components and the functionality therein”, see [13, SWS Function Inhibition Manager]. By this means, the FIM can significantly modify the behavior of the application software at run-time, e.g. in response to a sensor failure.

The conditions for deciding about inhibitions are derived from *diagnostic events*. If a *diagnostic event* that relates to a hardware sensor is reported as “failed”, then the FIM

can degrade the behavior of the application software to no longer rely on the sensor information that has become unreliable.

Consequently, the DiagnosticExtract needs to provide the basic mechanisms to create the definition of such inhibition rules that relate pieces of the application software (which are visible to the DiagnosticExtract as formally defined SwComponentTypes) to the definition of *diagnostic events*.

2.4.2 Model Fim configuration before Dem exists

The configuration of the FIM module and the DEM module are closely intertwined because the FIM uses the *diagnostic events* defined in the context of the DEM. This means that the FIM can only be configured if the DEM exists as a basis for creating this configuration.

Clearly, this contradicts the approach of a “decentralized configuration” of the diagnostics stack’s behavior. It may happen that the configuration of the FIM and the configuration of the DEM by means of the DiagnosticExtract are distributed over different organizations within a concrete vehicle project.

However, the organizations would not be able to make progress independently of each other because the work product of one organization is an immediate prerequisite for the other organization to begin work.

Therefore, one important use case for the DiagnosticExtract is the ability to configure the FIM module in the scope of the DiagnosticExtract **without needing immediate access to the corresponding configuration of the DEM module**.

In other words, a typical work-flow could look like this:

1. Create the FIM configuration on the basis of a collection of alias objects that represent the *diagnostic events* that are defined in the context of the DEM module within the configuration of the FIM and **in parallel** create the DEM configuration and thereby add definitions of the actual *diagnostic events*.
2. Ship the configuration of the DEM to the organization that is responsible for the creation of the configuration of the FIM.
3. Let this organization take care of resolving the alias *diagnostic events* by way of referring from the FIM configuration to the *diagnostic events* contained in the actual DEM configuration.

2.5 Configuration of J1939 Diagnostics

Within the System Extract, J1939 diagnostics is configured on a service level by the diagnostic messages DM01..DM60, which are handled by J1939 DCM.

The diagnostic content which is provided by the diagnostic messages is defined within the DiagnosticExtract separately for each J1939 function, and contributes to the configuration of the J1939 DCM as well as the parts of the DEM configuration that are relevant for J1939.

J1939 functions (known as Controller Application or CA within the [14, SAE J1939 Standard]) are identified by a NAME and an address (assigned at run-time) and are represented in AUTOSAR by the J1939NmNode.

2.5.1 Modeling of J1939 Diagnostic Aspects independent of the Deployment

The main use case for modeling J1939 diagnostics in the DiagnosticExtract is the possibility to create the diagnostic configuration relevant for a J1939 function, even if the underlying communication architecture is not (fully) defined and if no vehicle model (represented by a System) exists.

The J1939 diagnostics modeling is done during functional development before the System exists in which the functions are deployed. This approach represents a case of decentralized configuration of the diagnostic stack.

2.5.2 J1939 Diagnostic Content modeled in the Diagnostic Extract

The diagnostic content for J1939 consists of the following topics:

SPN The Suspect Parameter Number is used throughout the J1939 specification to identify measured values (physical signals) and commands, the communication signals to which these are mapped, and diagnostic events caused by these.

Signals Communication signals that are relevant for diagnostics are specifically annotated, and reported by the J1939 diagnostics.

DTCs J1939 DTCs consist of an SPN, identifying the source of the diagnostic event, and an FMI (Failure Mode Indicator), identifying the problem with this source, like a boundary exception. In addition, an event counter is contained in the DTC.

Freeze Frames contain measured values captured at the time a diagnostic event is reported as failed. J1939 supports two different kinds of freeze frames:

- **standardized freeze frames** (reported by DM04), which have a legislated layout
- **expanded freeze frames** (reported by DM24/DM25), which have a configurable layout

3 Conceptual Background

Chapter 1 has already given an overview of the intended way of using the DiagnosticExtract template and files. This chapter gives further background information on the overall concept behind the new format to create a better basis for understanding of the meta-model described in Chapter 4.2.

3.1 Definition of relevant Diagnostic Elements

[TPS_DEXT_01046] ECU configuration is not suitable to be exchanged between partners in an ECU development project ⌈The ECU configuration (EcuC) parameters defined by the AUTOSAR Software Specification (SWS) documents for Dem and Dcm are not suitable to be exchanged between partners in an ECU development project.

Besides proprietary ways of using the EcuC format, the main reason for EcuC parameters being inappropriate to be exchanged is their closeness to implementation (e.g. parameters such as buffer sizes).⌋

[TPS_DEXT_01047] Differences in the development processes for diagnostics at automotive OEMs and ECU suppliers ⌈Additionally, there are differences in the development processes for diagnostics at automotive OEMs and ECU suppliers resulting in different views on relevant diagnostic properties to be exchanged and different ways of deriving and defining them as diagnostic requirements.⌋

Therefore, the identification of all diagnostic properties and requirements as a superset from the companies' views forms the basis on which the DiagnosticExtract template has been defined.

3.2 Abstraction from EcuC Level

The DiagnosticExtract template does not only focus on relevant diagnostic properties and requirements but also - if required - lift them onto an appropriate abstraction level to make them meaningfully exchangeable (e.g. debouncing requirements that abstract from mapping on a concrete ECU).

However, for many EcuC parameters identified as relevant, an abstraction is not useful or not required and thus, these parameters are mapped 1:1 to equivalent elements of the DiagnosticExtract template.

[TPS_DEXT_01140] Values contained in DiagnosticExtract shall be taken for the derivation of basic software modules ⌈The values specified in a given DiagnosticExtract shall be taken for the derivation of basic software modules in the diagnostic stack.

This means that parameter values in the EcuC are created that could be subject to constraints, e.g. a valid interval.⌋

Therefore, model elements in the DiagnosticExtract shall be carefully checked against the constraints formulated in EcuC. However, this document does not contain specific constraints on a detailed basis.

3.3 Independence of Definition

With respect to development processes, the DiagnosticExtract format also enables more independence when defining requirements on diagnostic functionality than is possible with EcuC parameters. The approach of “decentralized configuration” is implemented in two main ways, described in the following subsections.

3.3.1 Enabling separation of elements over several physical files

Most elements of the DiagnosticExtract template can be split over several physical files. Therefore, parts of these elements (e.g. certain attributes) can be defined by, for example, an OEM and other parts of these elements by, for example, an ECU supplier.

3.3.2 Use of self-contained mapping elements

Many diagnostic requirements are established by mappings between diagnostic elements (e.g., DTC to DemEvent mapping). However, the “decentralized configuration” approach requires that these mappings can be flexibly defined at almost any time within the ECU development process and by any of the companies or roles involved.

Therefore, the DiagnosticExtract template defines self-contained mapping elements that have references to two (or potentially more) diagnostic elements to define a mapping.

The self-contained mapping elements can be created at any time after the diagnostic elements to be mapped together have been defined. A mapping element can also be created after only one diagnostic element has been defined, indicating that the mapping can be completed by defining the additional diagnostic element(s).

4 Diagnostic Resources

4.1 Overview

This chapter describes the diagnostic resources used to define the external behavior of the diagnostic stack. These resources (e.g. diagnostic services and diagnostic events) are defined independently of their use by other diagnostic resources or the application software.

The diagnostic resource model is described in five sections:

- Common meta-classes shared by the diagnostic services (which roughly correspond to the Dcm) and diagnostic event handling (which roughly corresponds to the Dem), see Chapter 4.2.
- Configuration of diagnostic services, see Chapter 4.3.
- Configuration of diagnostic event handling, see Chapter 4.4.
- Configuration of functional inhibition from a diagnostic perspective, see Chapter 4.5.
- Configuration of J1939 diagnostics, see Chapter 4.6.

In accordance with [15, TPS Generic Structure Template], the modeling of the `DiagnosticExtract` supports variant handling for selected model elements.

4.2 Common Meta Model Elements

4.2.1 Overview

This chapter highlights the formal modeling of some of the central parts of AUTOSAR diagnostics when it comes to configuration. There are some concepts widely used that need to be reflected in the meta-model.

The purpose of the `DiagnosticCommonElement` is to provide a common reference target for all kinds of diagnostic elements. This aspect is explained in more detail in Chapter 4.2.7.

The purpose of a `Data Identifier` (`DID`) is to associate a unique numerical value with a piece of data related to diagnostics. From the modeling point of view, this means that the modeling of the `Data Identifier` needs to provide an attribute that represents the numeric value as well as a relation to a `Data Element` representing a set of diagnostic data.

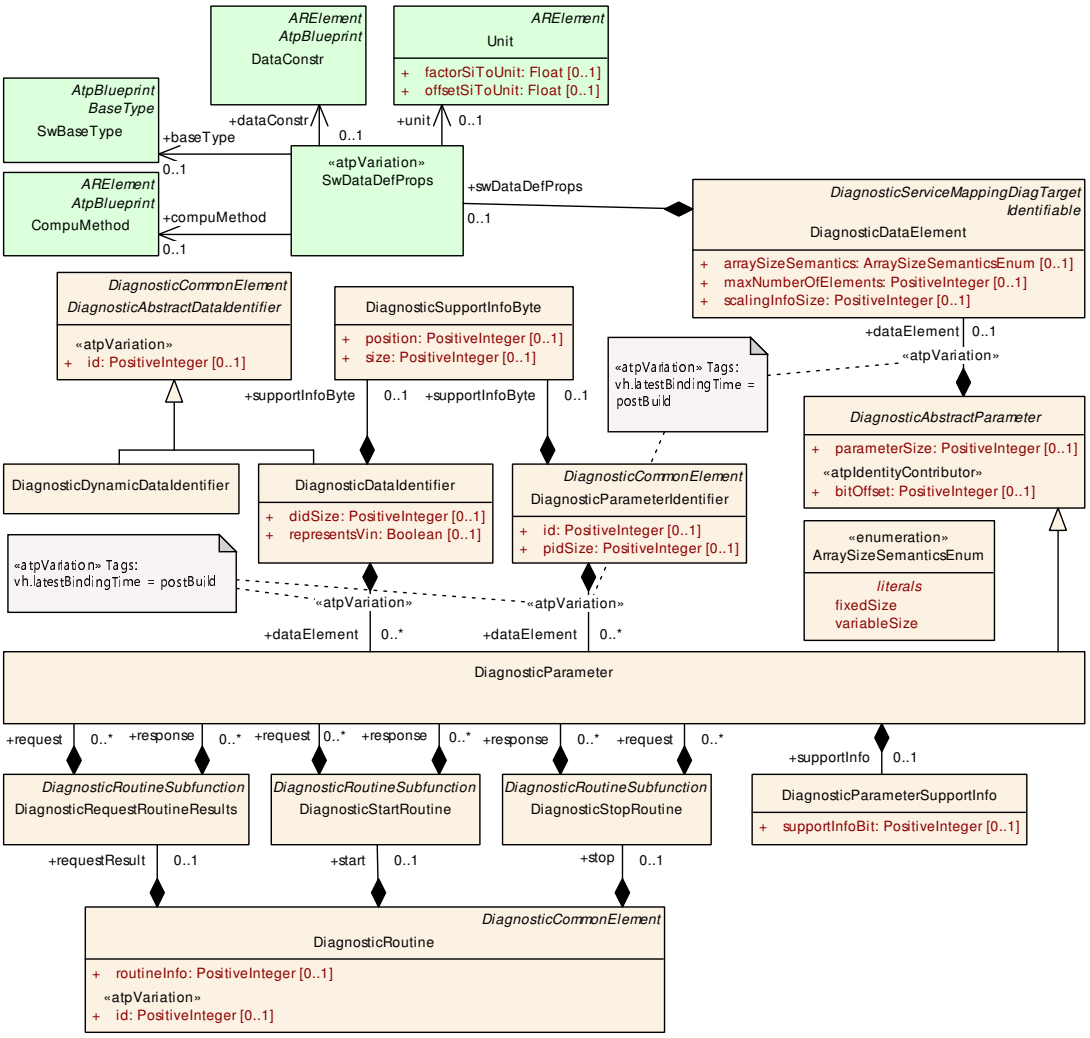

Figure 4.1: Common Diagnostic elements

4.2.2 Diagnostic Data Identifier

[TPS_DEXT_01000] AUTOSAR diagnostics supports two kinds of data identifiers
⌈In AUTOSAR, two kinds of data identifiers are supported:

- The `DiagnosticDataIdentifier` inherits from `DiagnosticAbstractDataIdentifier` and is used to define data identifiers fully known at configuration time. A `DiagnosticDataIdentifier` shall have at least 1 `dataElement`.
- The `DiagnosticDynamicDataIdentifier` inherits from `DiagnosticAbstractDataIdentifier` and is used to define data identifiers fully known **only** at run time. Consequently, there is **no formal means** to define `dataElement` at configuration time.

⌋

[constr_1792] Existence of DiagnosticDataIdentifier.dataElement

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticDataIdentifier, the aggregation of DiagnosticParameter in the role dataElement shall exist at least once.⌋

Class	**DiagnosticDataIdentifier**			
Note	This meta-class represents the ability to model a diagnostic data identifier (DID) that is fully specified regarding the payload at configuration-time. **Tags:** atp.recommendedPackage=DiagnosticDataIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAbstractDataIdentifier*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataElement	DiagnosticParameter	*	aggr	This is the dataElement associated with the Diagnostic DataIdentifier. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataElement, dataElement.bitOffset, data Element.ident.shortName, dataElement.variation Point.shortLabel vh.latestBindingTime=postBuild
didSize	PositiveInteger	0..1	attr	This attribute indicates the size in bytes of the Diagnostic DataIdentifier.
representsVin	Boolean	0..1	attr	This attributes indicates whether the specific Diagnostic DataIdentifier represents the vehicle identification.
supportInfoByte	DiagnosticSupportInfo Byte	0..1	aggr	This attribute represents the supported information associated with the DiagnosticDataIdentifier. **Tags:** atp.Splitkey=supportInfoByte

Table 4.1: DiagnosticDataIdentifier

Class	**DiagnosticDynamicDataIdentifier**			
Note	This meta-class represents the ability to define a diagnostic data identifier (DID) at run-time. **Tags:** atp.recommendedPackage=DiagnosticDataIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAbstractDataIdentifier*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.2: DiagnosticDynamicDataIdentifier

Class	***DiagnosticAbstractDataIdentifier*** (abstract)			
Note	This meta-class represents an abstract base class for the modeling of a diagnostic data identifier (DID).			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticDataIdentifier, DiagnosticDynamicDataIdentifier			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	***DiagnosticAbstractDataIdentifier*** (abstract)			
id	PositiveInteger	0..1	attr	This is the numerical identifier used to identify the DiagnosticAbstractDataIdentifier in the scope of diagnostic workflow **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table 4.3: DiagnosticAbstractDataIdentifier

[constr_1793] Existence of attribute `DiagnosticAbstractDataIdentifier.id`

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈For each `DiagnosticAbstractDataIdentifier`, attribute `id` shall exist.⌋

[TPS_DEXT_01072] Purpose of the attribute `DiagnosticDataIdentifier.representsVin` ⌈There is a use case for identifying a specific `DiagnosticDataIdentifier` that carries the so-called *vehicle identification number* (`VIN`).

It is therefore important to be able to formally indicate this characteristic. For this purpose the attribute `DiagnosticDataIdentifier.representsVin` is available.⌋

[constr_1324] Existence of `DiagnosticDataIdentifier.representsVin`

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈Within the context of a given `DiagnosticContributionSet`, the attribute `DiagnosticDataIdentifier.representsVin` shall have the value `true` for only a single `DiagnosticDataIdentifier`.⌋

Please note that the `VIN` is only relevant in the context of *diagnostics over IP* (`DoIP`). However, there is no constraint that bounds the validity of [constr_1324] to the existence of a `DiagnosticConnection` that is built on top of an IP stack.

If the attribute exists and IP is not used, then the meaning of the attribute is simply irrelevant. In any case, this situation should not be attributed to a misconfiguration.

4.2.3 Diagnostic Parameter

4.2.3.1 Overview

[TPS_DEXT_01169] Role of meta-class `DiagnosticParameter` ⌈The role of the `DiagnosticParameter` is to allocate a well-defined (i.e. start, length, content) slot in a diagnostic telegram, represented e.g. by a `DiagnosticDataIdentifier` or `DiagnosticRoutineSubfunction`.⌋

To facilitate the modeling of `DiagnosticParameter`, an abstract meta-class named `DiagnosticAbstractParameter` is defined to allow for making statements (e.g. [TPS_DEXT_01170], [TPS_DEXT_01171], and [TPS_DEXT_01174]) about all its subclasses uniformly instead of clarifying model semantics for each sub-class separately.

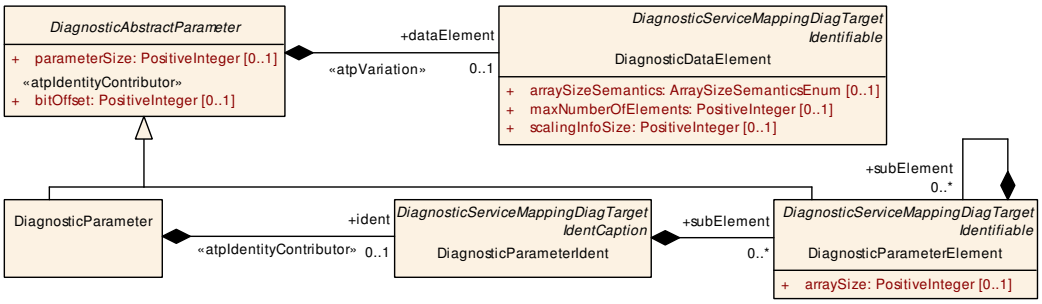

Figure 4.2: Modeling of `DiagnosticParameter`

[TPS_DEXT_01170] Semantics of meta-class `DiagnosticAbstractParameter`
⌈The meta-class `DiagnosticAbstractParameter` has been defined as a vessel to define attributes that are equally relevant for the sub-classes `DiagnosticParameter` and `DiagnosticParameterElement`.⌋

The modeling approach for the definition of a `DiagnosticParameter` is reflected in Figure 4.2.

Class	**DiagnosticParameter**			
Note	This meta-class represents the ability to describe information relevant for the execution of a specific diagnostic service, i.e. it can be taken to parameterize the service.			
Base	*ARObject*, *DiagnosticAbstractParameter*			
Aggregated by	DiagnosticDataIdentifier.dataElement, DiagnosticInfoType.dataElement, DiagnosticParameterIdentifier.dataElement, DiagnosticRequestRoutineResults.request, DiagnosticRequestRoutineResults.response, DiagnosticStartRoutine.request, DiagnosticStartRoutine.response, DiagnosticStopRoutine.request, DiagnosticStopRoutine.response			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ident	DiagnosticParameterIdent	0..1	aggr	The aggregation in the role ident provides the ability to make the DiagnosticAbstractParameter identifiable. From the semantical point of view, the AbstractDiagnosticParameter is considered a first-class Identifiable and therefore the aggregation in the role ident shall always exist (until it may be possible to let AbstractDiagnosticParameter directly inherit from Identifiable). **Stereotypes:** atpIdentityContributor
supportInfo	DiagnosticParameterSupportInfo	0..1	aggr	This attribute represents the ability to define which bit of the support info byte is representing this part of the PID. **Tags:** atp.Splitkey=supportInfo

Table 4.4: DiagnosticParameter

Class	**DiagnosticParameterElement**			
Note	This meta-class represents an element of a DiagnosticParameter if the DiagnosticParameter represents a structure.			
Base	*ARObject*, *DiagnosticAbstractParameter*, *DiagnosticServiceMappingDiagTarget*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticParameterElement.subElement, DiagnosticParameterIdent.subElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	DiagnosticParameterElement			
arraySize	PositiveInteger	0..1	attr	This attribute indicates that the enclosing Diagnostic ParameterElement represents an array and configures the array size in terms of the number of elements of the array.
subElement	DiagnosticParameter Element	*	aggr	This collection represents the sub-elements on the next lower level. **Tags:** atp.Splitkey=subElement, subElement.shortName, subElement.bitOffset

Table 4.5: DiagnosticParameterElement

Class	DiagnosticParameterIdent			
Note	This meta-class has been created to introduce the ability to become referenced into the meta-class AbstractDiagnosticParameter without breaking backwards compatibility.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *DiagnosticServiceMappingDiagTarget*, *Ident Caption*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, DiagnosticParameter.ident			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
subElement	DiagnosticParameter Element	*	aggr	This collection represents the subElements on the top level. **Tags:** atp.Splitkey=subElement, subElement.shortName, subElement.bitOffset

Table 4.6: DiagnosticParameterIdent

Class	***DiagnosticAbstractParameter*** (abstract)			
Note	This meta-class represents an abstract base class for modeling a diagnostic parameter.			
Base	*ARObject*			
Subclasses	DiagnosticParameter, DiagnosticParameterElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
bitOffset	PositiveInteger	0..1	attr	This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticData Identifier, DiagnosticParameterIdentifier, or Diagnostic RoutineSubfunction. **Stereotypes:** atpIdentityContributor **Tags:** atp.Status=candidate
dataElement	DiagnosticDataElement	0..1	aggr	This represents the related dataElement of the Diagnostic Parameter **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataElement, dataElement.shortName, data Element.variationPoint.shortLabel vh.latestBindingTime=postBuild
parameterSize	PositiveInteger	0..1	attr	This attribute allows for the specification of the parameter size. This information is relevant if there is a gap between one diagnostic parameter and the following diagnostic parameter (or the tail of the telegram). The unit is bit and the values shall be multiples of 8. **Tags:** atp.Status=candidate

Table 4.7: DiagnosticAbstractParameter

4.2.3.2 Size of a Diagnostic Parameter

[TPS_DEXT_01171] Size of a DiagnosticAbstractParameter ⌈The size of the slot can be described in two **mutually exclusive** ways:

- if the aggregation in the role DiagnosticAbstractParameter.dataElement exists, then the size of the slot is defined by the value of the attribute DiagnosticAbstractParameter.dataElement.swDataDefProps.baseType.baseTypeSize. The unit of attribute baseTypeSize is *bit*.
- if the aggregation DiagnosticAbstractParameter.dataElement does not exist, then the size of the slot is defined by the value of the attribute DiagnosticAbstractParameter.parameterSize. The unit of attribute parameterSize is *bit*.

⌋

[TPS_DEXT_01172] Semantics of the primitive attribute DiagnosticAbstractParameter.parameterSize ⌈The primitive attribute DiagnosticAbstractParameter.parameterSize can be used to indicate the size of the current DiagnosticAbstractParameter in the unit *bits*.

This information is relevant if the following DiagnosticAbstractParameter (or the end of the telegram) is not placed directly adjacent to the current DiagnosticAbstractParameter, i.e. gaps exist between two adjacent DiagnosticAbstractParameters.⌋

4.2.3.3 Start Position of a Diagnostic Parameter

[TPS_DEXT_01173] Start position of a DiagnosticAbstractParameter ⌈The slot allocated by a DiagnosticAbstractParameter is defined by means of its **starting position** (formalized as DiagnosticAbstractParameter.bitOffset, unit: *bit*) and the **slot's size**.⌋

[constr_1790] Existence of attribute DiagnosticAbstractParameter.bitOffset

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticParameter, attribute bitOffset shall exist.⌋

[constr_1470] Value of DiagnosticAbstractParameter.bitOffset

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈The value of DiagnosticAbstractParameter.bitOffset shall only be set to a multiple of 8.⌋

4.2.3.4 Content of a Diagnostic Parameter

[TPS_DEXT_01174] Content of a DiagnosticParameter ⌈A DiagnosticParameter defines its content (i.e. the information stored in the slot) in one of two possible ways:

- The DiagnosticParameter aggregates DiagnosticDataElement in the role dataElement.
- The DiagnosticParameter contains a sub-structure consisting of DiagnosticParameterElements by means of the aggregation DiagnosticParameter.ident.subElement. In this case, the definition of the actual content is delegated (recursively) to the sub-structure.

⌋

Please note that the existence of the aggregation in the role subElement can be used to form nested hierarchies that resemble **arrays** or **structures**, including a mixture of both.

Please note further that the ability to define a subElement with array-semantics is somehow redundant to the already available capability of DiagnosticDataElement, which can define an array of "primitive" content.

To reduce the level of redundancy, a subElement with array-semantics shall only be used for the definition of arrays with "non-primitive" content, i.e. the DiagnosticParameterElement that represents the array shall have further subElements (see [constr_10369]).

[constr_1791] Existence of attribute dataElement vs. parameterSize of meta-class DiagnosticParameter

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticParameter, exactly **one** of the attributes

- dataElement or
- parameterSize

shall exist.⌋

[TPS_DEXT_01175] Nested definition of a DiagnosticParameter ⌈It is possible to define a nested structure inside the definition of a DiagnosticParameter.

This way, it is possible to replicate the nested structure of an AutosarDataPrototype, such that either

- the entire structured AutosarDataPrototype or
- parts (i.e. DataPrototypes) of the structured data object

can be mapped to the respective locations in the definition of the structured DiagnosticParameter.

The definition of a structured DiagnosticParameter is done by aggregating DiagnosticParameterElement in the role ident.subElement.

A structured DiagnosticParameter can have a ident.subElement that itself represents a structure as well. For this purpose, the ident.subElement aggregates DiagnosticParameterElement in the role subElement.⌋

Please note: in AUTOSAR, the canonical naming approach for category LEAF would most likely boil down to the value VALUE. But in this specific case it is possible that the enclosed DiagnosticDataElement represents an **array** of primitive elements. Therefore, the usage of VALUE could be misleading in this case.

[constr_10369] Existence of attributes of DiagnosticParameterElement depending on the value of attribute category

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈

Value of category	Description	arraySize	subElement	dataElement
LEAF	The DiagnosticParameterElement represents a "leaf" element of a nested structure.	No	No	Yes
ARRAY	The DiagnosticParameterElement represents an array, i.e references to this DiagnosticParameterElement shall define a value for the attribute index. Arrays of "primitive" types are defined in the context of the DiagnosticDataElement	Yes	Yes, if dataElement does not exist	Yes, if subElement does not exist
STRUCTURE	The DiagnosticParameterElement represents a structure with one or more elements.	No	Yes	No

⌋

Please find examples for the definition of nested DiagnosticParameters in Chapter B.1.

4.2.3.5 Diagnostic Parameter as Reference Target

[TPS_DEXT_01176] DiagnosticParameter as a reference target ⌈The mapping of DataPrototypes inside a nested AutosarDataPrototype to a DiagnosticParameterElement requires the root DiagnosticParameter to define a shortName that becomes part of the reference in the ARXML serialization.

But DiagnosticParameter natively does not have the ability to define a shortName and therefore the aggregation of DiagnosticParameterIdent in the role ident is required to fill this gap.

For the case that the DiagnosticParameter directly defines a dataElement (i.e. the aggregation DiagnosticParameter.ident.subElement does not exist), the aggregation of DiagnosticParameterIdent in the role ident is **not** required.⌋

4.2.4 Diagnostic Data Element

The concept of the Data Element represents a piece of information decomposed from the data identified by a DID and exchanged between the diagnostic stack and, for example, a tester.

The nature of such a Data Element could be compared to the nature of an ISignal[1] and therefore the modeling of a Data Element by means of the meta-class DiagnosticDataElement aggregates SwDataDefProps in the role swDataDefProps in order to provide a reference to SwBaseType.

The aggregation of SwDataDefProps can also be used to refer to a DataConstr in order to specify a valid data range for the DiagnosticDataElement.

The definition of general blueprints (see [16, MOD General Blueprints]) in AUTOSAR offers a set of standardized SwBaseTypes for DEXT imports and exports (see /AUTOSAR/Diagnostic/BaseTypes_Blueprint) which allow the type's endianness on the network to be defined explicitly.

In contrast, the endianness of the platform base types is platform-dependent and therefore only defined in the context of an EcuInstance. This also means that in the context of two different EcuInstances, a SwBaseType with the same AUTOSAR Path can have different endianness.

Using the diagnostic SwBaseTypes avoids this potential problem.

[constr_1394] Value of DiagnosticDataElement.maxNumberOfElements depending on its existence

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈If the attribute DiagnosticDataElement.maxNumberOfElements exists, then its value shall be greater than 0.⌋

Class	**DiagnosticDataElement**			
Note	This meta-class represents the ability to describe a concrete piece of data to be taken into account for diagnostic purposes.			
Base	*ARObject*, *DiagnosticServiceMappingDiagTarget*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*DiagnosticAbstractParameter*.dataElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

[1] which represents the payload in "regular" bus communication

△

Class	**DiagnosticDataElement**			
arraySize Semantics	ArraySizeSemantics Enum	0..1	attr	This attribute controls the meaning of the value of the array size.
maxNumberOf Elements	PositiveInteger	0..1	attr	The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
scalingInfoSize	PositiveInteger	0..1	attr	Size in bytes of scaling information for the DiagnosticData Element if used with DiagnosticReadScalingDataBy Identifier
swDataDef Props	SwDataDefProps	0..1	aggr	This property allows to specify data definition properties in order to support the definition of e.g. computation formulae and data constraints. **Tags:** atp.Splitkey=swDataDefProps

Table 4.8: DiagnosticDataElement

[TPS_DEXT_01134] Definition of a `DiagnosticDataElement` used in the context of a DID obtained by sender-receiver communication ⌈If the `DiagnosticDataElement` is aggregated by a `DiagnosticParameter` that in turn is aggregated by a subclass of `DiagnosticAbstractDataIdentifier` and the `DiagnosticDataElement` is also referenced by a `DiagnosticServiceDataMapping` then the referenced `DiagnosticDataElement` (by way of `SwDataDefProps`) shall refer to a `SwBaseType` with attribute `baseTypeSize` set to **either** 8, 16, 32, or 64 and attribute `baseTypeEncoding` set to **either**

- `NONE` (for 8, 16, 32, or 64 bit) or
- `2C` (for 8, 16, 32, or 64 bit) or
- `IEEE754` (for 32 or 64 bit).

In this case it is possible to define the `DiagnosticDataElement` either as a scalar or as an array (see [TPS_DEXT_01001], [TPS_DEXT_01002]).⌋

[TPS_DEXT_01135] Definition of a `DiagnosticDataElement` used in the context of a DID obtained by client/server communication ⌈If the `DiagnosticDataElement` is aggregated by a `DiagnosticParameter` that in turn is aggregated by a subclass of `DiagnosticAbstractDataIdentifier` and the `DiagnosticDataElement` is also referenced by a `DiagnosticServiceSwMapping` then the referenced `DiagnosticDataElement` (by way of `SwDataDefProps`) shall refer to a `SwBaseType` with attribute `baseTypeSize` set to **either** 8, 16, 32, or 64 and attribute `baseTypeEncoding` set to **either**

- `NONE` (for 8, 16, 32, or 64 bit) or
- `2C` (for 8, 16, 32, or 64 bit) or
- `IEEE754` (for 32 or 64 bit).

In this case it is only possible to define the `DiagnosticDataElement` as an array (see [TPS_DEXT_01001], [TPS_DEXT_01002]). The ability to define a Variable-Size Array shall only be used for the last element of the `DID`.⌋

[TPS_DEXT_01136] Definition of a DiagnosticDataElement used in the context of a diagnostic routine ⌈If the DiagnosticDataElement is aggregated by a DiagnosticParameter that in turn is aggregated by either a DiagnosticStartRoutine, DiagnosticStopRoutine, or DiagnosticRequestRoutineResults then the DiagnosticDataElement (by way of SwDataDefProps) shall refer to a SwBaseType with attribute baseTypeSize to **either** 8, 16, 32, or 64 and attribute baseTypeEncoding set to **either**

- NONE (for 8, 16, 32, or 64 bit) or
- 2C (for 8, 16, 32, or 64 bit) or
- IEEE754 (for 32 or 64 bit).

In this case it is possible to define the DiagnosticDataElement either as a scalar or as an array (see [TPS_DEXT_01001], [TPS_DEXT_01002]). The ability to define a Variable-Size Array shall only be used for the last argument to the diagnostic routine.⌋

Enumeration	**ArraySizeSemanticsEnum**
Note	This enumeration provides options for how the number of elements in an array object shall be interpreted.
Aggregated by	ApplicationArrayElement.arraySizeSemantics, DiagnosticDataElement.arraySizeSemantics, ImplementationDataTypeElement.arraySizeSemantics, SwTextProps.arraySizeSemantics
Literal	***Description***
fixedSize	This means that the array object will always have a fixed number of elements. **Tags:** atp.EnumerationLiteralIndex=0
variableSize	This implies that the actual number of elements in the array object might vary at run-time. The configured size represents the maximum number of elements in the array object. **Tags:** atp.EnumerationLiteralIndex=1

Table 4.9: ArraySizeSemanticsEnum

The attribute DiagnosticDataIdentifier.didSize should be used if the size of the DID data is intended to be enforced.

[constr_1552] Restrictions for DiagnosticDataIdentifierSet referencing meta-class DiagnosticDataIdentifier

Imposition time: CP: IT_DiagDes

⌈If a DiagnosticDataIdentifier is referenced by a DiagnosticDataIdentifierSet, then the DiagnosticDataIdentifier shall fulfill **all of the following conditions**:

- The DiagnosticDataIdentifier does **not** aggregate any DiagnosticParameter which in turn aggregates (via the aggregation of DiagnosticParameterIdent) a DiagnosticParameterElement.
- Gaps in between individual elements (i.e. under consideration of the beginning of an individual DiagnosticDataElement, as indicated by DiagnosticParameter.bitOffset, and the length of the aggregated DiagnosticDataElement) or at the end of the DiagnosticDataIdentifier (as indicated by attribute DiagnosticDataIdentifier.didSize) shall **not** exist.

The individual DiagnosticDataElement contained (via the aggregation of DiagnosticParameter in the role dataElement) in the DiagnosticDataIdentifier shall satisfy **one of the following conditions**:

- The DiagnosticDataElement does **not** define the attribute maxNumberOfElements at all.
- The modeling of attribute DiagnosticDataElement.arraySizeSemantics shall follow the description in [TPS_DEXT_01001].

⌋

[constr_10368] Restriction regarding the reference in the role DiagnosticDataIdentifierSet.diagnosticDataIdentifier

Imposition time: CP: IT_DiagDes

⌈A DiagnosticDataIdentifier that is referenced in the role DiagnosticDataIdentifierSet.diagnosticDataIdentifier shall not aggregate in the role dataElement a DiagnosticParameter that aggregates in the role ident a DiagnosticParameterIdent that in turn aggregates in the role subElement a DiagnosticParameterElement.⌋

Rationale for the existence of [constr_10368]: in this context, there is no valid use case for defining nested DiagnosticParameters.

[TPS_DEXT_01138] Applicability of DiagnosticDataIdentifier.supportInfoByte ⌈The attribute DiagnosticDataIdentifier.supportInfoByte may exist if the value of DiagnosticDataIdentifier.id is **in the range 0xF400-0xF4FF**.

A typical case for the existence of DiagnosticDataIdentifier.supportInfoByte is the understanding that the DiagnosticDataIdentifier is relevant for OBD and the DiagnosticDataIdentifier.id is in the designated range.⌋

[constr_1472] Existence of DiagnosticDataIdentifier.supportInfoByte

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈The attribute DiagnosticDataIdentifier.supportInfoByte **shall not exist** if the value of DiagnosticDataIdentifier.id is **outside the range 0xF400-0xF4FF**.⌋

4.2.4.1 Usage of SwDataDefProps

Please note that the definitions of properties like computation methods[2], limits[3], or units[4] of diagnostic data elements are based on shared resources of the AUTOSAR meta-model, specifically, through aggregation of meta-class `SwDataDefProps`.

[constr_1325] Allowed attributes of `SwDataDefProps` for `DiagnosticDataElement.swDataDefProps`

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈

Attributes of `SwDataDefProps`	`DiagnosticDataElement.swDataDefProps`
`additionalNativeTypeQualifier`	
`annotation`	
`baseType.baseTypeDefinition.baseTypeEncoding`	D
`baseType.baseTypeDefinition.baseTypeSize`	D
`baseType.baseTypeDefinition.byteOrder`	D
`baseType.baseTypeDefinition.memAlignment`	
`baseType.baseTypeDefinition.nativeDeclaration`	
`compuMethod`	D
`dataConstr`	D
`displayFormat`	D
`displayPresentation`	
`implementationDataType`	
`invalidValue`	
`swAddrMethod`	
`swAlignment`	
`swBitRepresentation`	
`swCalibrationAccess`	
`swCalprmAxisSet`	
`swComparisonVariable`	
`swDataDependency`	
`swImplPolicy`	
`swIntendedResolution`	
`swInterpolationMethod`	
`swIsVirtual`	
`swPointerTargetProps`	
`swRecordLayout`	
`swRefreshTiming`	
`swTextProps`	
`swValueBlockSize`	
`unit`	D
`valueAxisDataType`	

⌋

[2]formalized as `CompuMethod` in AUTOSAR
[3]formalized as `DataConstr` in AUTOSAR
[4]formalized as `Unit` in AUTOSAR

Legend for Table inside [constr_1325]

D Define the attribute.

empty cell Attribute is **not applicable** for usage in the scope of this element.

This meta-class contributes a wealth of possible properties related to the definition of data in general and, in this case, diagnostics in particular.

However, it is important to understand that SwDataDefProps is so expressive and versatile that its applicability needs to be constrained (in this specific case, see [constr_1325]) for specific deployments according to the requirements that originate from the semantics of the piece of data that is decorated by SwDataDefProps.

Please note that, in comparison to similar constraints appearing in other AUTOSAR documents (e.g. [8, TPS Software Component Template]), [constr_1325] intentionally goes into more detail regarding the applicability of the attributes of SwBaseType. This is in contrast to similar constraints contained in, e.g. the specification of the [8, TPS Software Component Template].

The attributes of SwBaseType are considered to be of **paramount importance for the definition of the semantics** of the enclosing DiagnosticDataElement and thus the emphasis is justified.

Class	«atpVariation» **SwDataDefProps**
Note	This class is a collection of properties relevant for data objects under various aspects. One could consider this class as a "pattern of inheritance by aggregation". The properties can be applied to all objects of all classes in which SwDataDefProps is aggregated. Note that not all of the attributes or associated elements are useful all of the time. Hence, the process definition (e.g. expressed with an OCL or a Document Control Instance MSR-DCI) has the task of implementing limitations. SwDataDefProps covers various aspects: • Structure of the data element for calibration use cases: is it a single value, a curve, or a map, but also the recordLayouts which specify how such elements are mapped/converted to the DataTypes in the programming language (or in AUTOSAR). This is mainly expressed by properties like swRecordLayout and swCalprmAxisSet • Implementation aspects, mainly expressed by swImplPolicy, swAddrMethod, baseType, implementationDataType and additionalNativeTypeQualifier • Access policy for the MCD system, mainly expressed by swCalibrationAccess • Semantics of the data element, mainly expressed by compuMethod and/or unit, dataConstr, invalid Value • Code generation policy provided by swRecordLayout **Tags:** vh.latestBindingTime=codeGenerationTime
Base	*ARObject*

▽

△

Class	«atpVariation» **SwDataDefProps**			
Aggregated by	*AutosarDataType*.swDataDefProps, CompositeNetworkRepresentation.networkRepresentation, CppImplementationDataTypeElement.swDataDefProps, *DataPrototype*.swDataDefProps, DataPrototypeTransformationProps.networkRepresentationProps, DiagnosticDataElement.swDataDefProps, DiagnosticEnvDataElementCondition.swDataDefProps, DiagnosticExtendedDataRecordElement.swDataDefProps, DiagnosticSovdPrimitiveContentElement.swDataDefProps, DltArgumentProps.networkRep, FlatInstanceDescriptor.swDataDefProps, ImplementationDataTypeElement.swDataDefProps, InstantiationDataDefProps.swDataDefProps, ISignal.networkRepresentationProps, McDataInstance.resultingProperties, ParameterAccess.swDataDefProps, PerInstanceMemory.swDataDefProps, *ReceiverComSpec*.networkRepresentation, SecurityEventContextDataElement.networkRepresentation, *SenderComSpec*.networkRepresentation, SomeipDataPrototypeTransformationProps.networkRepresentation, SwPointerTargetProps.swDataDefProps, SwServiceArg.swDataDefProps, SwSystemconst.swDataDefProps, SystemSignal.physicalProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
additionalNativeTypeQualifier	NativeDeclarationString	0..1	attr	This attribute is used to declare native qualifiers of the programming language which can neither be deduced from the baseType (e.g. because the data object describes a pointer) nor from other more abstract attributes. Examples are qualifiers like "volatile", "strict" or "enum" of the C-language. All such declarations have to be put into one string. **Tags:** xml.sequenceOffset=235
annotation	Annotation	*	aggr	This aggregation allows to add annotations (yellow pads ...) related to the current data object. **Tags:** atp.Splitkey=annotation xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false
baseType	SwBaseType	0..1	ref	Base type associated with the containing data object. **Tags:** atp.Splitkey=baseType xml.sequenceOffset=50
compuMethod	CompuMethod	0..1	ref	Computation method associated with the semantics of this data object. **Tags:** atp.Splitkey=compuMethod xml.sequenceOffset=180
dataConstr	DataConstr	0..1	ref	Data constraint for this data object. **Tags:** atp.Splitkey=dataConstr xml.sequenceOffset=190
displayFormat	DisplayFormatString	0..1	attr	This property describes how a number is to be rendered e.g. in documents or in a measurement and calibration system. **Tags:** xml.sequenceOffset=210
displayPresentation	DisplayPresentationEnum	0..1	attr	This attribute controls the presentation of the related data for measurement and calibration tools.

▽

△

Class	«atpVariation» **SwDataDefProps**			
implementation DataType	AbstractImplementation DataType	0..1	ref	This association denotes the ImplementationDataType of a data declaration via its aggregated SwDataDefProps. It is used whenever a data declaration is not directly referring to a base type. Especially • redefinition of an ImplementationDataType via a "typedef" to another ImplementationDatatype • the target type of a pointer (see SwPointerTarget Props), if it does not refer to a base type directly • the data type of an array or record element within an ImplementationDataType, if it does not refer to a base type directly • the data type of an SwServiceArg, if it does not refer to a base type directly **Tags:** atp.Splitkey=implementationDataType xml.sequenceOffset=215
invalidValue	ValueSpecification	0..1	aggr	Optional value to express invalidity of the actual data element. **Tags:** atp.Splitkey=invalidValue xml.sequenceOffset=255
stepSize	Float	0..1	attr	This attribute can be used to define a value which is added to or subtracted from the value of a DataPrototype when using up/down keys while calibrating.
swAddrMethod	SwAddrMethod	0..1	ref	Addressing method related to this data object. Via an association to the same SwAddrMethod it can be specified that several DataPrototypes shall be located in the same memory without already specifying the memory section itself. **Tags:** atp.Splitkey=swAddrMethod xml.sequenceOffset=30
swAlignment	AlignmentType	0..1	attr	The attribute describes the intended typical alignment of the DataPrototype. If the attribute is not defined the alignment is determined by the swBaseType size and the memoryAllocationKeywordPolicy of the referenced Sw AddrMethod. **Tags:** xml.sequenceOffset=33
swBit Representation	SwBitRepresentation	0..1	aggr	Description of the binary representation in case of a bit variable. **Tags:** atp.Splitkey=swBitRepresentation xml.sequenceOffset=60
swCalibration Access	SwCalibrationAccess Enum	0..1	attr	Specifies the read or write access by MCD tools for this data object. **Tags:** xml.sequenceOffset=70
swCalprmAxis Set	SwCalprmAxisSet	0..1	aggr	This specifies the properties of the axes in case of a curve or map etc. This is mainly applicable to calibration parameters. **Tags:** atp.Splitkey=swCalprmAxisSet xml.sequenceOffset=90

▽

△

Class	«atpVariation» **SwDataDefProps**			
swComparison Variable	SwVariableRefProxy	*	aggr	Variables used for comparison in an MCD process. **Tags:** atp.Splitkey=swComparisonVariable xml.sequenceOffset=170 xml.typeElement=false
swData Dependency	SwDataDependency	0..1	aggr	Describes how the value of the data object has to be calculated from the value of another data object (by the MCD system). **Tags:** atp.Splitkey=swDataDependency xml.sequenceOffset=200
swHostVariable	SwVariableRefProxy	0..1	aggr	Contains a reference to a variable which serves as a host-variable for a bit variable. Only applicable to bit objects. **Tags:** atp.Splitkey=swHostVariable xml.sequenceOffset=220 xml.typeElement=false
swImplPolicy	SwImplPolicyEnum	0..1	attr	Implementation policy for this data object. **Tags:** xml.sequenceOffset=230
swIntended Resolution	Numerical	0..1	attr	The purpose of this element is to describe the requested quantization of data objects early on in the design process. The resolution ultimately occurs via the conversion formula present (compuMethod), which specifies the transition from the physical world to the standardized world (and vice-versa) (here, "the slope per bit" is present implicitly in the conversion formula). In the case of a development phase without a fixed conversion formula, a pre-specification can occur through swIntendedResolution. The resolution is specified in the physical domain according to the property "unit". **Tags:** xml.sequenceOffset=240
swInterpolation Method	Identifier	0..1	attr	This is a keyword identifying the mathematical method to be applied for interpolation. The keyword needs to be related to the interpolation routine which needs to be invoked. **Tags:** xml.sequenceOffset=250
swIsVirtual	Boolean	0..1	attr	This element distinguishes virtual objects. Virtual objects do not appear in the memory, their derivation is much more dependent on other objects and hence they shall have a swDataDependency . **Tags:** xml.sequenceOffset=260
swPointerTarget Props	SwPointerTargetProps	0..1	aggr	Specifies that the containing data object is a pointer to another data object. **Tags:** atp.Splitkey=swPointerTargetProps xml.sequenceOffset=280
swRecord Layout	SwRecordLayout	0..1	ref	Record layout for this data object. **Tags:** atp.Splitkey=swRecordLayout xml.sequenceOffset=290

▽

△

Class	«atpVariation» **SwDataDefProps**			
swRefresh Timing	MultidimensionalTime	0..1	aggr	This element specifies the frequency in which the object involved shall be or is called or calculated. This timing can be collected from the task in which write access processes to the variable run. But this cannot be done by the MCD system. So this attribute can be used in an early phase to express the desired refresh timing and later on to specify the real refresh timing. **Tags:** atp.Splitkey=swRefreshTiming xml.sequenceOffset=300
swTextProps	SwTextProps	0..1	aggr	the specific properties if the data object is a text object. **Tags:** atp.Splitkey=swTextProps xml.sequenceOffset=120
swValueBlock Size	Numerical	0..1	attr	This represents the size of a Value Block **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=80
swValueBlock SizeMult (ordered)	Numerical	*	attr	This attribute is used to specify the dimensions of a value block (VAL_BLK) for the case that that value block has more than one dimension. The dimensions given in this attribute are ordered such that the first entry represents the first dimension, the second entry represents the second dimension, and so on. For one-dimensional value blocks the attribute swValue BlockSize shall be used and this attribute shall not exist. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
unit	Unit	0..1	ref	Physical unit associated with the semantics of this data object. This attribute applies if no compuMethod is specified. If both units (this as well as via compuMethod) are specified the units shall be compatible. **Tags:** atp.Splitkey=unit xml.sequenceOffset=350
valueAxisData Type	ApplicationPrimitive DataType	0..1	ref	The referenced ApplicationPrimitiveDataType represents the primitive data type of the value axis within a compound primitive (e.g. curve, map). It supersedes CompuMethod, Unit, and BaseType. **Tags:** atp.Splitkey=valueAxisDataType xml.sequenceOffset=355

Table 4.10: SwDataDefProps

4.2.5 Definition of Arrays on the level of DiagnosticDataElement

There are several use cases for a `DiagnosticDataElement` that actually represents an array of information. In some cases the array size is static and will not change at run-time, and in some cases the array size needs to change at run-time to fulfill the intended purpose.

[TPS_DEXT_01001] Definition of a fixed-sized array ⌈A DiagnosticDataElement shall be interpreted as a **fixed-size array** if **all** of the following conditions apply:

1. The attribute DiagnosticDataElement.maxNumberOfElements exists.
2. The value of the attribute DiagnosticDataElement.maxNumberOfElements is set to a value > 0.
3. The value of DiagnosticDataElement.arraySizeSemantics either does not exist or is set to ArraySizeSemanticsEnum.fixedSize.

⌋

[TPS_DEXT_01002] Definition of a variable-sized array ⌈A DiagnosticDataElement shall be interpreted as a **variable-size array** if **all** of the following conditions apply:

1. The attribute DiagnosticDataElement.maxNumberOfElements exists.
2. The value of the attribute DiagnosticDataElement.maxNumberOfElements is set to a value > 0.
3. The value of DiagnosticDataElement.arraySizeSemantics is set to ArraySizeSemanticsEnum.variableSize.

The value of DiagnosticDataElement.maxNumberOfElements shall be considered the maximum array size in terms of the number of elements.⌋

[constr_1326] Existence of a variable-sized array

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈The value of the attribute DiagnosticDataElement.arraySizeSemantics **shall not** be set to ArraySizeSemanticsEnum.variableSize if the respective DiagnosticDataElement is referenced from a DiagnosticServiceDataMapping.⌋

4.2.5.1 Definition of textual Strings

DiagnosticDataElement can be used to model a **textual string** that shall be sent to or received from the ECU by the tester.

In this case it will be necessary to define the **intended encoding** such that the part of the software on the ECU that produces or consumes the string on the ECU can be checked (after the establishment of a DiagnosticMapping) for a matching encoding specification.

The encoding in the scope of the DiagnosticDataElement can be defined using the attribute DiagnosticDataElement.swDataDefProps.baseType.baseTypeDefinition.baseTypeEncoding.

4.2.6 Textual Documentation

A Data Identifier also usually comes with some textual description that explains the meaning of the Data Identifier in short form. This ability is available via the inheritance from Identifiable, in particular by means of the attributes desc and/or introduction (see Figure 4.3).

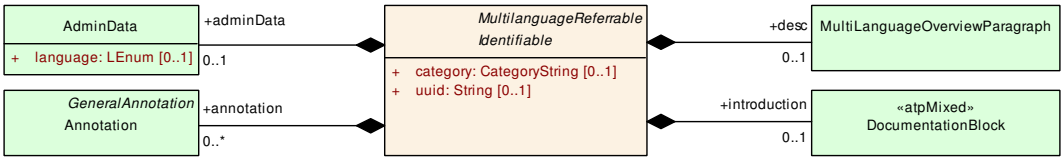

Figure 4.3: Definition of a textual description by means of desc and introduction

This also means that the ability to add some form of textual description is widely usable in the scope of the DiagnosticExtract.

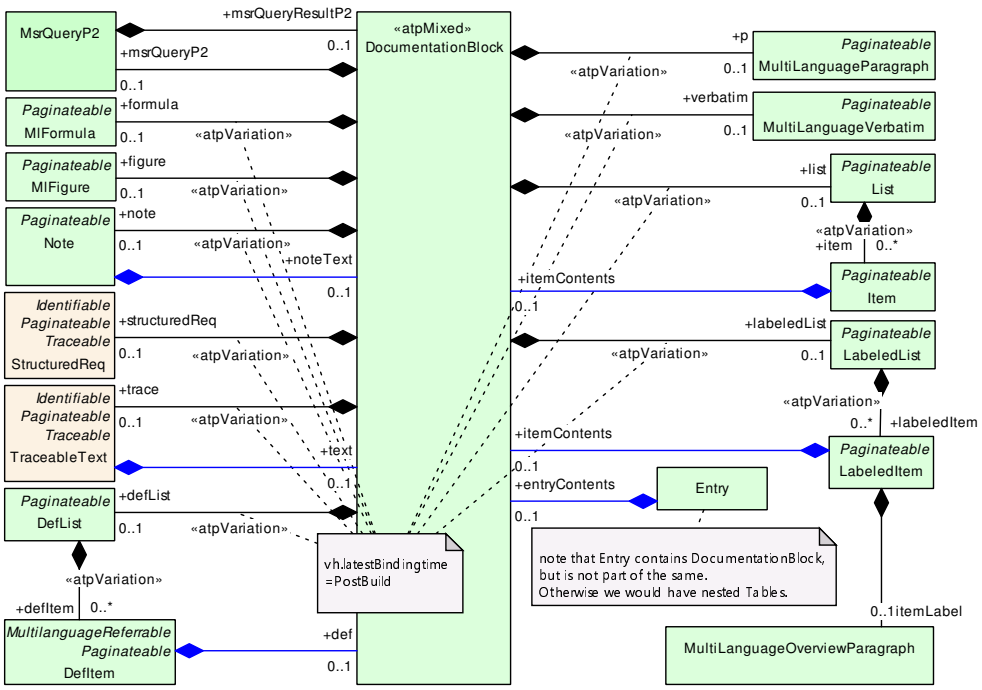

Figure 4.4: Details of the modeling of DocumentationBlock

Many meta-classes are derived from e.g. DiagnosticCommonElement (which inherits from Identifiable) or directly from Identifiable and therefore qualify for the described form of documentation.

In other words, the technology described in Figure 4.3 and Figure 4.4 is not limited to DiagnosticDataElement but has a much wider applicability in the context of the DiagnosticExtract.

The details regarding the specification of textual content that goes along with a given diagnostics element are detailed in Figure 4.4.

In fact, DocumentationBlock provides a very sophisticated ability to define structured text that may consist e.g. of multiple paragraphs (formalized by meta-class MultiLanguageOverviewParagraph aggregated in the role p).

In addition to the ability to attach structured text, it is also possible to use the annotation (see Figure 4.3) to add short annotations (comparable to the usage of sticky notes) to diagnostic elements.

Class	«atpMixed» **DocumentationBlock**			
Note	This class represents a documentation block. It is made of basic text structure elements which can be displayed in a table cell.			
Base	*ARObject*			
Aggregated by	ApplicabilityInfo.remark, AUTOSAR.introduction, BlueprintGenerator.introduction, *BlueprintPolicy* *Modifiable*.blueprintDerivationGuide, *DataMapping*.introduction, DefItem.def, *Describable*.introduction, EcucAddInfoParamValue.value, EcuResourceEstimation.introduction, Entry.entryContents, Frame Mapping.introduction, *GeneralAnnotation*.annotationText, *Identifiable*.introduction, IPdu Mapping.introduction, ISignalMapping.introduction, Item.itemContents, LabeledItem.itemContents, Life CycleInfo.remark, *MappingConstraint*.introduction, MsrQueryP2.msrQueryResultP2, Note.noteText, *ScheduleTableEntry*.introduction, *SignalPathConstraint*.introduction, StructuredReq.conflicts, Structured Req.dependencies, StructuredReq.description, StructuredReq.rationale, StructuredReq.remark, StructuredReq.supportingMaterial, StructuredReq.useCase, SwAxisType.swGenericAxisDesc, Topic Content.blockLevelContent, TraceableText.text, VariationPoint.blueprintCondition			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
defList	DefList	0..1	aggr	This represents a definition list in the documentation block. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=40
figure	MIFigure	0..1	aggr	This represents a figure in the documentation block. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=70
formula	MIFormula	0..1	aggr	This is a formula in the definition block. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=60
labeledList	LabeledList	0..1	aggr	This represents a labeled list. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=50
list	List	0..1	aggr	This represents numbered or unnumbered list. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=30
msrQueryP2	MsrQueryP2	0..1	aggr	This represents automatically contributed contents provided by an msrquery in the context of Documentation Block.



△

Class	«atpMixed» **DocumentationBlock**			
note	Note	0..1	aggr	This represents a note in the text flow. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=80
p	MultiLanguage Paragraph	0..1	aggr	This is one particular paragraph. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=10
structuredReq	StructuredReq	0..1	aggr	This aggregation supports structured requirements embedded in a documentation block. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=100
trace	TraceableText	0..1	aggr	This represents traceable text in the documentation block. This allows to specify requirements/constraints in any documentation block. The kind of the trace is specified in the category. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=90
verbatim	MultiLanguageVerbatim	0..1	aggr	This represents one particular verbatim text. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=20

Table 4.11: DocumentationBlock

Class	**MultiLanguageOverviewParagraph**			
Note	This is the content of a multilingual paragraph in an overview item.			
Base	*ARObject*			
Aggregated by	Caption.desc, CompuScale.desc, *Describable*.desc, *Identifiable*.desc, LabeledItem.itemLabel, Modification.change, Modification.reason, SdgCaption.desc, SwRecordLayoutGroup.desc, SwRecord LayoutV.desc, VariationPoint.desc			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
l2	LOverviewParagraph	1..*	aggr	This represents the text in one particular language. **Tags:** atp.Splitkey=l2 xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table 4.12: MultiLanguageOverviewParagraph

4.2.7 Diagnostic Contribution

[TPS_DEXT_01003] **DiagnosticContributionSet is the central part of the DiagnosticExtract** ⌈The central part of the formalization of the concept of the DiagnosticExtract is the DiagnosticContributionSet.

To some extent, it fulfills a similar role as the System (see [5, TPS System Template]) in the communication domain.⌋

The DiagnosticContributionSet references to DiagnosticCommonElement and by this means the actual definition of the extent of diagnostic contribution takes place.

In other words, the actual extent of a given contribution is created by the collection of DiagnosticCommonElements referenced by the DiagnosticContributionSet.

[TPS_DEXT_01004] **DiagnosticContributionSet defines the scope of the DiagnosticExtract** ⌈The DiagnosticContributionSet has the ability to define the scope of the given DiagnosticExtract. This means that the DiagnosticContributionSet represents the DiagnosticExtract for the rest of the AUTOSAR model.

In other words, the DiagnosticContributionSet assumes a similar crucial role as the System in that it defines the extent to which a given description of diagnostic behavior is valid.

Any downstream processing of the description of diagnostic behavior shall take the scope of information to be processed from the definition of the DiagnosticContributionSet in its role as the root element of the description of diagnostic behavior.

During different stages of the design methodology, the scope of the DiagnosticContributionSet may vary between the scope of an entire system down to the contribution of a specific tier-1 supplier to a much bigger context.⌋

[TPS_DEXT_01055] **Standardized values of DiagnosticContributionSet.category** ⌈The scope of the DiagnosticContributionSet, on the other hand, is determined by the value of its category. The following values are predefined by AUTOSAR:

DIAGNOSTICS_ABSTRACT_SYSTEM_DESCRIPTION this DiagnosticContributionSet represents a more or less high-level definition that can be taken as a template for creating concrete DiagnosticContributionSets of category DIAGNOSTICS_SYSTEM_EXTRACT or DIAGNOSTICS_ECU_EXTRACT

DIAGNOSTICS_SYSTEM_EXTRACT the scope of this DiagnosticContributionSet consists of several EcuInstances.

DIAGNOSTICS_ECU_EXTRACT the scope of this DiagnosticContributionSet consists of a **single** EcuInstance. A DiagnosticContributionSet of category DIAGNOSTICS_ECU_EXTRACT shall be taken as the basis for the derivation of service software-components of the diagnostic stack.

⌋

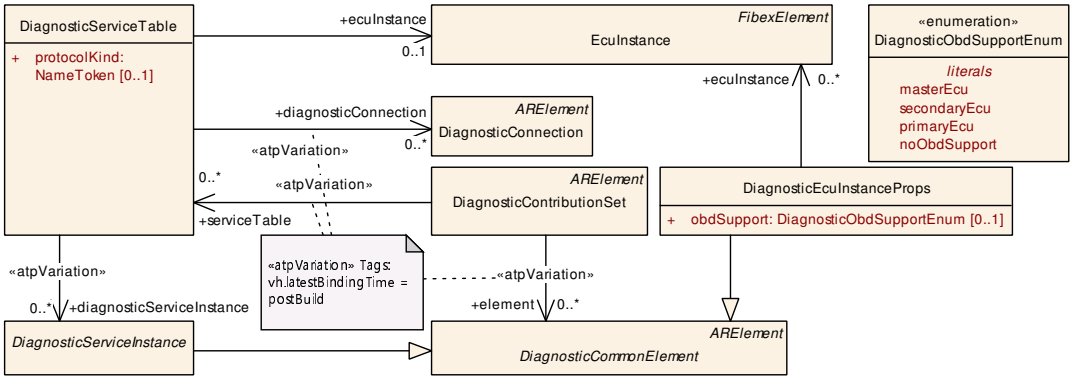

Figure 4.5: Diagnostic Contribution

[constr_1327] Multiplicity of DiagnosticEcuInstanceProps.ecuInstance

Imposition time: CP: IT_DiagDes

⌈The multiplicity of DiagnosticEcuInstanceProps.ecuInstance shall be limited to 1 and the enclosing DiagnosticContributionSet shall only refer to at most one DiagnosticEcuInstanceProps **if** the enclosing DiagnosticContributionSet is of category DIAGNOSTICS_ECU_EXTRACT.⌋

[constr_1328] Consistency of DiagnosticEcuInstanceProps.ecuInstance and DiagnosticServiceTable.ecuInstance

Imposition time: CP: IT_DiagDes

⌈Each DiagnosticServiceTable referenced by any given DiagnosticContributionSet in the role serviceTable shall define a reference in the role DiagnosticServiceTable.ecuInstance to an EcuInstance that is also referenced in the role DiagnosticEcuInstanceProps.ecuInstance by a DiagnosticEcuInstanceProps referenced by the mentioned DiagnosticContributionSet **if** the respective DiagnosticContributionSet is of category DIAGNOSTICS_ECU_EXTRACT.⌋

Please note that [constr_1328] resolves an intentional redundancy in the meta-model. Both DiagnosticContributionSet and DiagnosticServiceTable are able to refer to EcuInstance with the idea that both DiagnosticContributionSet and DiagnosticServiceTable can be modeled independently of each other.

Of course, once the DiagnosticContributionSet and DiagnosticServiceTable are integrated in the same context (in particular by establishing the reference DiagnosticContributionSet.serviceTable) the individual references to the applicable EcuInstances need to be consistent.

[TPS_DEXT_01005] DiagnosticContributionSet can exist independently
⌈The DiagnosticContributionSet has been modeled as an ARElement so that its instances can exist independently of the existence of context-providing model-elements inside a given ARPackage.⌋

[TPS_DEXT_01005] elaborates on an important aspect that makes the DiagnosticExtract independent of the existence of a context. For example, it would have been possible to aggregate DiagnosticContributionSet somewhere, e.g. at System.

This kind of modeling intentionally puts DiagnosticContributionSet on the same level as e.g. System, as far as model granularity is concerned.

The purpose of the DiagnosticContributionSet is to refer to all DiagnosticCommonElements that are relevant in the context of the DiagnosticContributionSet.

It is not sufficient that a given DiagnosticCommonElement is referenced by another DiagnosticCommonElement but is not referenced by the same DiagnosticContributionSet that also references the referencing DiagnosticCommonElement.

Class	**DiagnosticContributionSet**			
Note	This meta-class represents a root node of a diagnostic extract. It bundles a given set of diagnostic model elements. The granularity of the DiagonsticContributionSet is arbitrary in order to support the aspect of decentralized configuration, i.e. different contributors can come up with an own DiagnosticContribution Set. **Tags:** atp.recommendedPackage=DiagnosticContributionSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
common Properties	DiagnosticCommon Props	0..1	aggr	This attribute represents a collection of diagnostic properties that are shared among the entire Diagnostic ContributionSet. **Tags:** atp.Splitkey=commonProperties
element	DiagnosticCommon Element	*	ref	This represents a DiagnosticCommonElement considered in the context of the DiagnosticContributionSet **Stereotypes:** atpVariation **Tags:** atp.Splitkey=element, element.diagnosticCommon Element, element.variationPoint.shortLabel vh.latestBindingTime=postBuild
serviceTable	DiagnosticServiceTable	*	ref	This represents the collection of DiagnosticServiceTables to be considered in the scope of this Diagnostic ContributionSet. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=serviceTable, serviceTable.diagnosticService Table, serviceTable.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 4.13: DiagnosticContributionSet

Class	**DiagnosticCommonElement** (abstract)			
Note	This meta-class represents a common base class for all diagnostic elements. It does not contribute any specific functionality other than the ability to become the target of a reference.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	*DiagnosticAbstractAliasEvent*, *DiagnosticAbstractDataIdentifier*, DiagnosticAccessPermission, DiagnosticAging, DiagnosticAuthRole, *DiagnosticCondition*, *DiagnosticConditionGroup*, DiagnosticCustomServiceClass, DiagnosticDataIdentifierSet, DiagnosticEcuInstanceProps, DiagnosticEnvironmentalCondition, DiagnosticEvent, DiagnosticExtendedDataRecord, DiagnosticFimEventGroup, DiagnosticFreezeFrame, DiagnosticFunctionIdentifier, DiagnosticFunctionIdentifierInhibit, DiagnosticIndicator, DiagnosticInfoType, DiagnosticIumpr, DiagnosticIumprDenominatorGroup, DiagnosticIumprGroup, DiagnosticJ1939ExpandedFreezeFrame, DiagnosticJ1939FreezeFrame, DiagnosticJ1939Node, DiagnosticJ1939Spn, *DiagnosticMapping*, DiagnosticMeasurementIdentifier, *DiagnosticMemoryDestination*, DiagnosticMemoryIdentifier, DiagnosticOperationCycle, DiagnosticParameterIdentifier, DiagnosticPowertrainFreezeFrame, DiagnosticProtocol, DiagnosticRoutine, DiagnosticSecurityLevel, *DiagnosticServiceClass*, *DiagnosticServiceInstance*, DiagnosticServiceTable, DiagnosticSession, DiagnosticTestResult, DiagnosticTestRoutineIdentifier, *DiagnosticTroubleCode*, DiagnosticTroubleCodeGroup, DiagnosticTroubleCodeProps			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.14: DiagnosticCommonElement

[constr_1745] Indirect reference to `DiagnosticCommonElement`

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈If a `DiagnosticCommonElement` is referenced from within the context of another `DiagnosticCommonElement` and the referencing `DiagnosticCommonElement` is in turn referenced by a `DiagnosticContributionSet` in the role `element` then the referenced `DiagnosticCommonElement` shall also be referenced by the same `DiagnosticContributionSet`.⌋

Example 4.1

A `DiagnosticDataIdentifier` that is referenced by a `DiagnosticDataIdentifierSet` that is referenced by a `DiagnosticContributionSet` shall also be referenced by that `DiagnosticContributionSet`. This scenario is depicted in Figure 4.6.

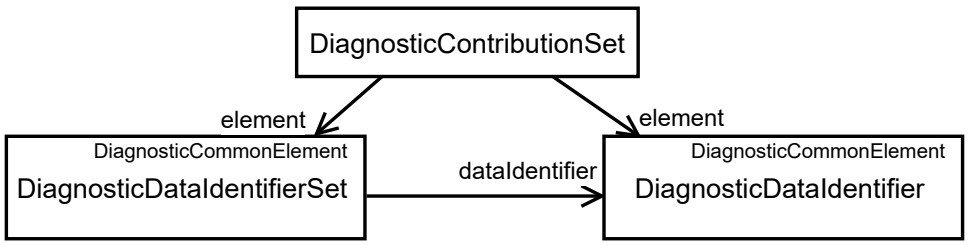

Figure 4.6: Visualization of example scenario for the indirect reference to `DiagnosticCommonElement`

4.2.8 Diagnostic Protocol

[TPS_DEXT_01124] Semantics of meta-class `DiagnosticProtocol` ⌈The meta-class `DiagnosticProtocol` can be used to describe the usage of different diagnostic protocols and their priority.⌋

Class	**DiagnosticProtocol**			
Note	This meta-class represents the ability to define a diagnostic protocol. **Tags:** atp.recommendedPackage=DiagnosticProtocols			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnosticConnection	DiagnosticConnection	*	ref	This represents the collection of applicable Diagnostic Connections for this DiagnosticProtocol. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=diagnosticConnection, diagnosticConnection.diagnosticConnection, diagnosticConnection.variationPoint.shortLabel vh.latestBindingTime=postBuild
priority	PositiveInteger	0..1	attr	This represents the priority of the diagnostic protocol in comparison to other diagnostic protocols. Lower numeric values represent higher protocol priority: • 0 - Highest protocol priority • 255 - Lowest protocol priority **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
protocolKind	NameToken	0..1	attr	This identifies the applicable protocol.
sendRespPendOnTransToBoot	Boolean	0..1	attr	The purpose of this attribute is to define whether or not the ECU should send a NRC 0x78 (response pending) before transitioning to the bootloader (in this case the attribute shall be set to "true") or if the transition shall be initiated without sending NRC 0x78 (in this case the attribute shall be set to "false"). **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
serviceTable	DiagnosticServiceTable	0..1	ref	This represents the service table applicable for the given diagnostic protocol. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=serviceTable, serviceTable.diagnosticServiceTable, serviceTable.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 4.15: DiagnosticProtocol

[constr_1794] Existence of attribute `DiagnosticProtocol.priority`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticProtocol`, attribute `priority` shall exist.⌋

[constr_1795] Existence of attribute `DiagnosticProtocol.protocolKind`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticProtocol`, attribute `protocolKind` shall exist.⌋

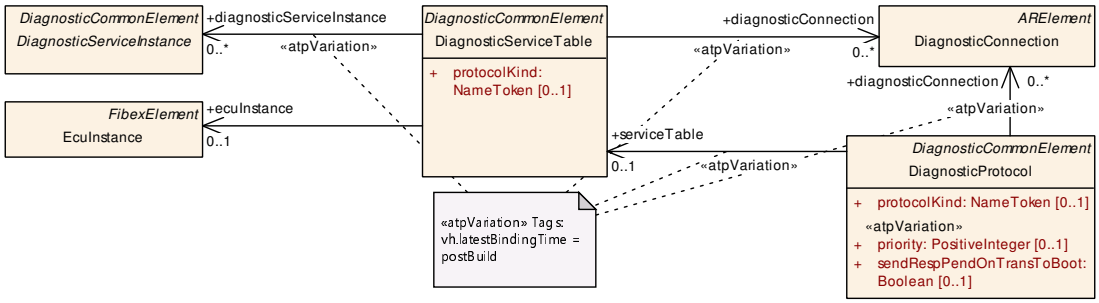

Figure 4.7: Modeling of `DiagnosticProtocol`

Each `DiagnosticProtocol` refers to at most one `DiagnosticServiceTable` and to a collection of `DiagnosticConnection`s.

Please note that both `DiagnosticServiceTable` and `DiagnosticProtocol` have an attribute named `protocolKind`.

Class	**DiagnosticServiceTable**			
Note	This meta-class represents a model of a diagnostic service table, i.e. the UDS services applicable for a given ECU. **Tags:** atp.recommendedPackage=DiagnosticServiceTables			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnostic Connection	DiagnosticConnection	*	ref	This represents the DiagnosticConnection that is taken for handling the data transmission for the enclosing DiagnosticServiceTable. It is possible to refer to more than one diagnostic Connections in order to support more than one diagnostic tester. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=diagnosticConnection, diagnostic Connection.diagnosticConnection, diagnostic Connection.variationPoint.shortLabel vh.latestBindingTime=postBuild
diagnostic ServiceInstance	DiagnosticService Instance	*	ref	This represents the collection of DiagnosticService Instances to be considered in the scope of this Diagnostic ServiceTable. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=diagnosticServiceInstance, diagnostic ServiceInstance.diagnosticServiceInstance, diagnostic ServiceInstance.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	**DiagnosticServiceTable**			
ecuInstance	EcuInstance	0..1	ref	This represents the applicable EcuInstance for this DiagnosticServiceTable. **Tags:** atp.Splitkey=ecuInstance
protocolKind	NameToken	0..1	attr	This identifies the applicable protocol.

Table 4.16: DiagnosticServiceTable

[constr_10573] Existence of attribute DiagnosticServiceTable.diagnosticServiceInstance

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticServiceTable, attribute diagnosticServiceInstance shall exist.⌋

[constr_1797] Existence of attribute DiagnosticServiceTable.protocolKind

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticServiceTable, attribute protocolKind shall exist.⌋

The attribute DiagnosticServiceTable.protocolKind shall be used to define the applicability of a DiagnosticServiceTable for a given protocol before the formal definition of the protocol even exists.

In other words, the attribute gives the designer of the DiagnosticServiceTable a means to express an intention about the usage of the DiagnosticServiceTable.

The attribute DiagnosticServiceTable.protocolKind, on the other hand, is used to define the actual nature of the DiagnosticProtocol.

By means of DiagnosticProtocol.serviceTable both "ends" of this aspect are put together and therefore it is reasonable to express a constraint about the value of attribute protocolKind.

[constr_1405] Value of DiagnosticProtocol.serviceTable vs. DiagnosticServiceTable.protocolKind

Imposition time: CP: IT_DiagDes

⌈If the reference DiagnosticProtocol.serviceTable exists, then the value of DiagnosticProtocol.protocolKind shall be identical to the value of DiagnosticServiceTable.protocolKind.⌋

[TPS_DEXT_01006] The role of DiagnosticServiceTables ⌈The existence of a DiagnosticServiceTable creates a formal relation between a collection of DiagnosticServiceInstances and the DiagnosticConnection that formalizes a *conduit* for specific pairs of diagnostic request and response messages taken to transmit the diagnostic service invocations from a tester to the applicable instance of the AUTOSAR diagnostic stack and convey the response of the diagnostic stack back to the tester.

In particular, this means that a DiagnosticServiceTable describes the set of DiagnosticServiceInstances that are available via DiagnosticConnection, which is finally a request message to address a diagnostic service to an ECU and a response message to be used by the ECU to respond to the service.⌋

4.2.8.1 Implications on the Workflow to create a Diagnostic Extract

Please note that the definition of the DiagnosticConnection happens out of the scope of the definition of the diagnostic extract; it is typically provided by the system-wide communication design.

This means that, in order to create references in the roles DiagnosticProtocol.diagnosticConnection and DiagnosticServiceTable.diagnosticConnection it is necessary to know the qualified shortName path of the DiagnosticConnection to be referenced.

In a typical project, there are certain resources that become stable in very early stages because the whole project depends on the definition of these resources. The definition of the DiagnosticConnection is one of these resources. Therefore, it is reasonable to assume that the definition of the DiagnosticConnection is available at the time when the diagnostic extract is created.

On top of that, many OEMs use deterministic hierarchies of ARPackage so that the qualified shortName path of a DiagnosticConnection can be guessed/determined in advance and can be used to create the references in the roles DiagnosticProtocol.diagnosticConnection and DiagnosticServiceTable.diagnosticConnection.

Class	**DiagnosticConnection**			
Note	DiagnosticConncection that is used to describe the relationship between several TP connections. **Tags:** atp.recommendedPackage=DiagnosticConnections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
functional Request	TpConnectionIdent	*	ref	Reference to functional request messages. **Tags:** atp.Splitkey=functionalRequest
periodic ResponseUudt	PduTriggering	*	ref	Reference to UUDT responses. **Tags:** atp.Splitkey=periodicResponseUudt
physical Request	TpConnectionIdent	0..1	ref	Reference to a physical request message. **Tags:** atp.Splitkey=physicalRequest
response	TpConnectionIdent	0..1	ref	In the vast majority of cases a response is required. However, there are also cases where providing the response is not possible and/or not allowed. **Tags:** atp.Splitkey=response

Table 4.17: DiagnosticConnection

Class	**TpConnectionIdent**			
Note	This meta-class is created to add the ability to become the target of a reference to the non-Referrable Tp Connection.			
Base	*ARObject*, *Referrable*			
Aggregated by	*TpConnection*.ident			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table 4.18: TpConnectionIdent

Example 4.2 shows an exemplary definition of a `DiagnosticServiceTable`.

Example 4.2

$14 - GroupOfDTC: 0xFFFFFF

$19 - Subfunction: $02, Subfunction $06

$22 - DataID: 0x1111, DataID: 0x2222

$2E - DataID: 0x1111, DataID: 0x2222

$2F - IO-ID:0x3333

[TPS_DEXT_01091] Relation between a `DiagnosticServiceTable` and one or more `DiagnosticConnection`s ⌈In principle, the relation between a `DiagnosticServiceTable` and one or more `DiagnosticConnection`s can be established in two possible ways:

- by means of the reference in the role `DiagnosticServiceTable.diagnosticConnection`. This way, the concept of protocols and their priorities relative to each other is **not** considered.
- by means of the references in the roles `DiagnosticProtocol.diagnosticConnection` and `DiagnosticProtocol.serviceTable`. This way, the existence of several protocols and their priorities **is positively considered**.

⌋

Please note that alternatives mentioned in [TPS_DEXT_01091] may or may not be handled as alternatives in actual projects. It may be possible that in a first modeling step protocols are ignored entirely and therefore the respective relations are created by means of `DiagnosticServiceTable.diagnosticConnection`.

Later in time and as the project progresses, protocols may become a thing and are consequently introduced in the model. In response to this change, the relation in question is now created by means of the references `DiagnosticProtocol.diagnosticConnection` and `DiagnosticProtocol.serviceTable`.

However, the existing relation created by means of the reference `DiagnosticServiceTable.diagnosticConnection` is not necessarily required to be removed. It may just as well continue to exist. However, in this case an obvious consistency rule as described in [constr_1406] applies.

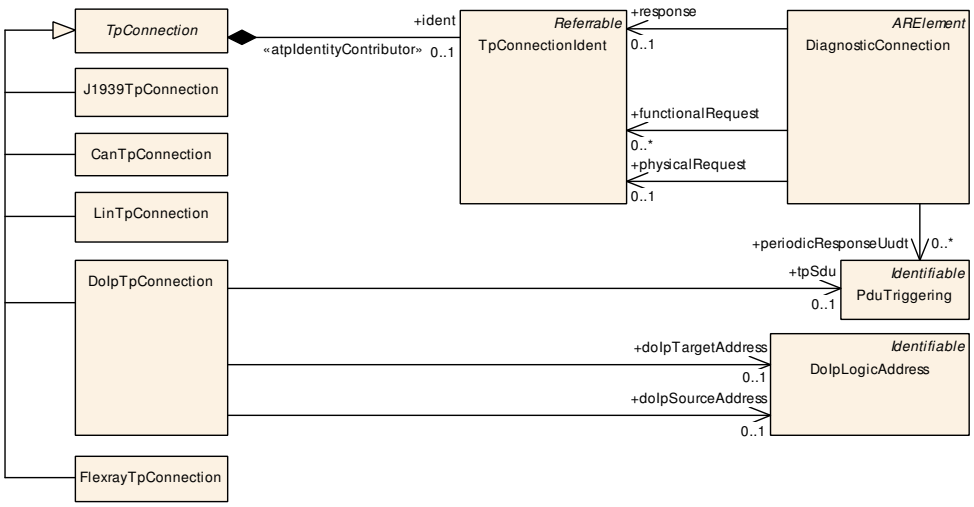

Figure 4.8: Modeling of `DiagnosticConnection`

[constr_1406] Meta-class `DiagnosticServiceTable.diagnosticConnection` vs. meta-class `DiagnosticProtocol.diagnosticConnection`

Imposition time: CP: `IT_DiagDes`

⌈If a `DiagnosticServiceTable` exists that fulfills the following conditions:

- reference `DiagnosticServiceTable.diagnosticConnection` exists
- the `DiagnosticServiceTable` is referenced by means of `DiagnosticProtocol.serviceTable`

then all of the `DiagnosticConnection`s referenced by means of `DiagnosticServiceTable.diagnosticConnection` shall also be referenced in the role `diagnosticConnection` from a `DiagnosticProtocol` that in turn references the respective `DiagnosticServiceTable` in the role `DiagnosticProtocol.serviceTable`.⌋

4.2.9 Diagnostic Common Properties

[TPS_DEXT_01007] Common properties of a `DiagnosticExtract` ⌈There are some properties of a `DiagnosticExtract` that are shared among all elements of the `DiagnosticExtract`. These properties are modeled by means of the meta-class `DiagnosticCommonProps`.⌋

[constr_10665] Existence of role `DiagnosticCommonProps.authenticationTimeout`

Imposition time: CP: `IT_DiagDes`, AP: `IT_DiagDes`

⌈If the `DiagnosticContributionSet` that aggregates a `DiagnosticCommonProps` also references a `DiagnosticAuthentication` in the role `element`, then attribute `DiagnosticCommonProps.authenticationTimeout` shall exist.⌋

[TPS_DEXT_01008] `DiagnosticContributionSet` defines the scope for the application of the common diagnostic properties ⌈`DiagnosticContributionSet` aggregates `DiagnosticCommonProps` and by this means defines the scope for the application of the common diagnostic properties.⌋

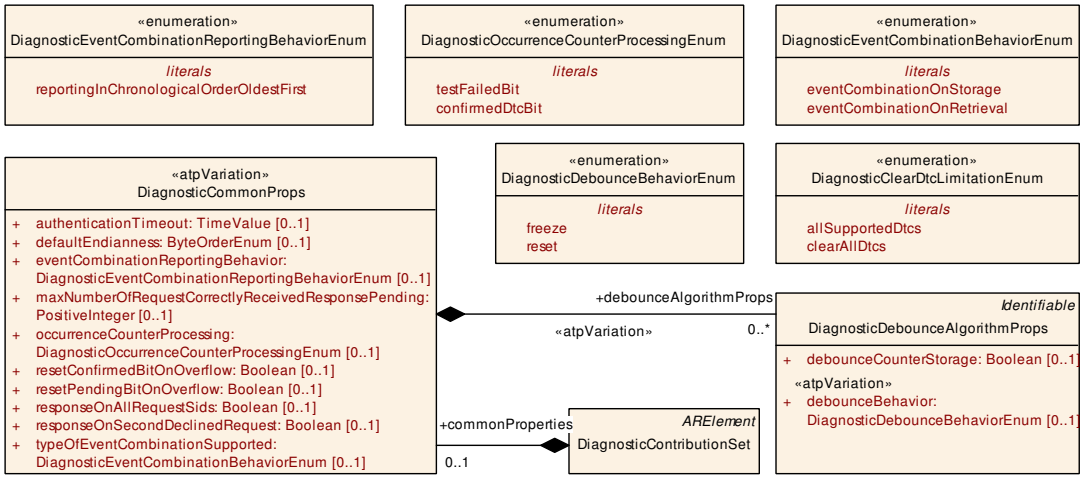

Figure 4.9: Common Diagnostic Properties

[constr_10042] Existence of the attribute `DiagnosticCommonProps.defaultEndianness`

Imposition time: CP: `IT_DiagDes`, AP: `IT_DiagDes`

⌈One of the following conditions shall be fulfilled:

- `DiagnosticCommonProps.defaultEndianness` exists.
- The primitive attribute `DiagnosticParameter.dataElement.swDataDefProps.baseType.baseTypeDefinition.baseTypeEncoding` exists for **all** `DiagnosticParameter`s defined in the context of the `DiagnosticContributionSet`.

⌋

[constr_10043] Existence of the primitive attribute `DiagnosticCommonProps.resetConfirmedBitOnOverflow`

Imposition time: CP: `IT_DiagDes`, AP: `IT_DiagDes`

⌈Attribute `DiagnosticCommonProps.resetConfirmedBitOnOverflow` shall exist.⌋

[constr_10044] Existence of primitive attribute DiagnosticCommonProps.occurrenceCounterProcessing

Imposition time: CP: IT_DiagDes

⌈If, in the context of a DiagnosticContributionSet, a DiagnosticEdrDataProviderMapping exists where attribute DiagnosticEdrDataProviderMapping.dataProvider is set to the value DEM_OCCCTR, then attribute DiagnosticCommonProps.occurrenceCounterProcessing shall exist.⌋

The following primitive attributes of meta-class DiagnosticCommonProps are optional:

- maxNumberOfRequestCorrectlyReceivedResponsePending
- responseOnAllRequestSids
- responseOnSecondDeclinedRequest

[constr_10089] Existence of the attribute DiagnosticCommonProps.eventCombinationReportingBehavior

Imposition time: CP: IT_DiagDes

⌈Attribute DiagnosticCommonProps.eventCombinationReportingBehavior is always optional and shall be set to the value DiagnosticEventCombinationReportingBehaviorEnum.reportingInChronologicalOrderOldestFirst only if the attribute DiagnosticCommonProps.typeOfEventCombinationSupported is set to the value DiagnosticEventCombinationBehaviorEnum.eventCombinationOnRetrieval.

If it is missing, then the reporting order is not specified.⌋

Class	«atpVariation» **DiagnosticCommonProps**			
Note	This meta-class aggregates a number of common properties that are shared among a diagnostic extract. **Tags:** vh.latestBindingTime=codeGenerationTime			
Base	*ARObject*			
Aggregated by	DiagnosticContributionSet.commonProperties			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
authentication Timeout	TimeValue	0..1	attr	This attribute defines the time (in seconds) that the authentication state is maintained in default-session if there is no communication from the authenticated client.
debounce AlgorithmProps	DiagnosticDebounce AlgorithmProps	*	aggr	Defines the used debounce algorithms relevant in the context of the enclosing DiagnosticCommonProps. Usually, there is a variety of debouncing algorithms to take into account and therefore the multiplicity of this aggregation is set to 0..*. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=debounceAlgorithmProps, debounce AlgorithmProps.shortName, debounceAlgorithm Props.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	«atpVariation» **DiagnosticCommonProps**			
default Endianness	ByteOrderEnum	0..1	attr	Defines the default endianness of the data belonging to a DID or RID which is applicable if the DiagnosticData Element does not define the endianness via the swData DefProps.baseType attribute.
event Combination Reporting Behavior	DiagnosticEvent CombinationReporting BehaviorEnum	0..1	attr	In case of EventCombination on Retrieval, this attribute specifies if a specific order of reporting is to be maintained.
maxNumberOf Request Correctly Received Response Pending	PositiveInteger	0..1	attr	Maximum number of negative responses with response code 0x78 (requestCorrectlyReceived-ResponsePending) allowed per request. DCM will send a negative response with response code 0x10 (generalReject), in case the limit value gets reached. Value 0xFF means that no limit number of NRC 0x78 response apply.
occurrence Counter Processing	DiagnosticOccurrence CounterProcessing Enum	0..1	attr	This attribute defines the consideration of the fault confirmation process for the occurrence counter.
resetConfirmed BitOnOverflow	Boolean	0..1	attr	This attribute defines whether the confirmed bit is reset or not while an event memory entry will be displaced or could not acquire an entry in the event memory.
resetPendingBit OnOverflow	Boolean	0..1	attr	This attribute defines whether the pending bit is reset or not while an event memory entry will be displaced or could not acquire an entry in the event memory. In order to be compliant to ISO 14229-1 [17], this parameter needs to be set to "false".
responseOnAll RequestSids	Boolean	0..1	attr	If set to FALSE the DCM will not respond to diagnostic request that contains a service ID which is in the range from 0x40 to 0x7F or in the range from 0xC0 to 0xFF (Response IDs).
responseOn Second Declined Request	Boolean	0..1	attr	Defines the reaction upon a second request (ClientB) that can not be processed (e.g. due to priority assessment). TRUE: when the second request (Client B) can not be processed, it shall be answered with NRC21 BusyRepeat Request. FALSE: when the second request (Client B) can not be processed, it shall not be responded.
typeOfEvent Combination Supported	DiagnosticEvent CombinationBehavior Enum	0..1	attr	Select type of Event Combination support.

Table 4.19: DiagnosticCommonProps

[constr_10419] Existence of the attribute DiagnosticCommonProps.resetPendingBitOnOverflow

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈Attribute DiagnosticCommonProps.resetPendingBitOnOverflow shall exist.⌋

Enumeration	**DiagnosticOccurrenceCounterProcessingEnum**
Note	The occurrence counter triggering types.
Aggregated by	DiagnosticCommonProps.occurrenceCounterProcessing
Literal	*Description*

▽

△

Enumeration	**DiagnosticOccurrenceCounterProcessingEnum**
confirmedDtcBit	The occurrence counter is incremented when TestFailed bit transitions from 0 to 1 if the fault confirmation was successful (ConfirmedDTC bit is already set). **Tags:** atp.EnumerationLiteralIndex=0
testFailedBit	The occurrence counter is incremented when TestFailed bit transitions from 0 to 1 (and the fault confirmation is not considered). **Tags:** atp.EnumerationLiteralIndex=1

Table 4.20: DiagnosticOccurrenceCounterProcessingEnum

Enumeration	**DiagnosticTypeOfDtcSupportedEnum**
Note	Supported Dtc Types
Aggregated by	DiagnosticMemoryDestinationPrimary.typeOfDtcSupported
Literal	***Description***
iso11992_4	ISO11992-4 DTC format **Tags:** atp.EnumerationLiteralIndex=0 xml.name=ISO-11992--4
iso14229_1	ISO14229-1 DTC format (3 byte format) **Tags:** atp.EnumerationLiteralIndex=1 xml.name=ISO-14229--1
iso15031_6	ISO 15031-6 DTC format (2 byte format) = SAE J2012-DA_DTCFormat_00 (2 Byte format) **Tags:** atp.EnumerationLiteralIndex=2 xml.name=ISO-15031--6
saeJ1939_73	SAEJ1939-73 DTC format **Tags:** atp.EnumerationLiteralIndex=3 xml.name=SAE-J-1939--73
saeJ2012_da	SAE_J2012-DA_DTCFormat_00 (3 byte format) **Tags:** atp.EnumerationLiteralIndex=4 xml.name=SAE-J-2012--DA

Table 4.21: DiagnosticTypeOfDtcSupportedEnum

Enumeration	**ByteOrderEnum**
Note	When more than one byte is stored in the memory the order of those bytes may differ depending on the architecture of the processing unit. If the least significant byte is stored at the lowest address, this architecture is called little endian and otherwise it is called big endian. ByteOrder is very important in case of communication between different PUs or ECUs.
Aggregated by	ApSomeipTransformationProps.byteOrder, BaseTypeDirectDefinition.byteOrder, DiagnosticCommonProps.defaultEndianness, ISignalToIPduMapping.packingByteOrder, MultiplexedIPdu.selectorFieldByteOrder, PduToFrameMapping.packingByteOrder, SegmentPosition.segmentByteOrder, SOMEIPTransformationDescription.byteOrder, System.containerIPduHeaderByteOrder
Literal	***Description***
mostSignificantByteFirst	Most significant byte shall come at the lowest address (also known as BigEndian or as Motorola-Format) **Tags:** atp.EnumerationLiteralIndex=0

▽

△

Enumeration	**ByteOrderEnum**
mostSignificantByte Last	Most significant byte shall come highest address (also known as LittleEndian or as Intel-Format) **Tags:** atp.EnumerationLiteralIndex=1
opaque	For opaque data endianness conversion has to be configured to Opaque. See AUTOSAR COM Specification for more details. **Tags:** atp.EnumerationLiteralIndex=2

Table 4.22: ByteOrderEnum

Enumeration	**DiagnosticEventCombinationBehaviorEnum**
Note	Select type of Event Combination support
Aggregated by	DiagnosticCommonProps.typeOfEventCombinationSupported
Literal	***Description***
eventCombination OnRetrieval	Event combination on retrieval is used to combine events. For each event an individual event memory entry is created, while reporting the data via UDS, the data is combined. **Tags:** atp.EnumerationLiteralIndex=1
eventCombination OnStorage	Event combination on storage is used to combine events. Only one memory entry exists for each DTC which is also reported via UDS. **Tags:** atp.EnumerationLiteralIndex=0

Table 4.23: DiagnosticEventCombinationBehaviorEnum

Enumeration	**DiagnosticEventCombinationReportingBehaviorEnum**
Note	Select reporting format of events. Applicable only for Event Combination on Retrieval.
Aggregated by	DiagnosticCommonProps.eventCombinationReportingBehavior
Literal	***Description***
reportingIn ChronologicalOrder OldestFirst	The reporting order for event combination on retrieval is the chronological storage order of the events **Tags:** atp.EnumerationLiteralIndex=1

Table 4.24: DiagnosticEventCombinationReportingBehaviorEnum

4.3 Diagnostic Services

4.3.1 Introduction

The meta-model for the diagnostic services according to AUTOSAR, to a large degree, takes over aspects of the description of diagnostic services according to the definition of *Unified Diagnostic Services* (UDS) as of [17, ISO 14229].

4.3.2 Diagnostic Service Instance vs. Diagnostic Service Class

When it comes to diagnostic services, the meta-model distinguishes between the concept of a diagnostic service *instance* vs. the concept of a diagnostic service *class*.

As the terminology suggests, the diagnostic service *instance* (formalized as meta-class DiagnosticServiceInstance) implements a concrete use of a diagnostic service

in a given context whereas the purpose of the diagnostic service *class* (formalized as DiagnosticServiceClass) is to provide properties that are shared among all existing diagnostic service *instance*s in the model.

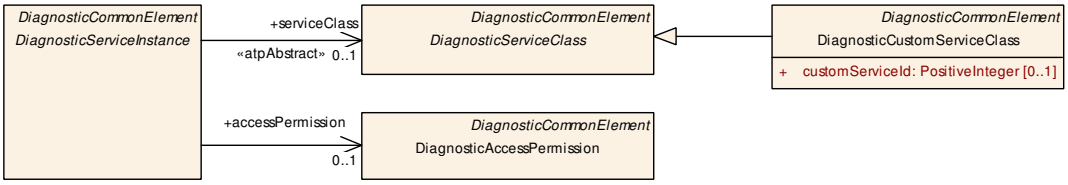

Figure 4.10: Common service elements

[constr_1329] Existence of concrete sub-classes of DiagnosticServiceClass in the context created by a DiagnosticContributionSet

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈One of the following mutually exclusive conditions shall apply for the existence of any concrete sub-class of DiagnosticServiceClass in the context created by a DiagnosticContributionSet:

- The subclass of DiagnosticServiceClass (except DiagnosticCustomServiceClass) shall only appear once in the context created by a DiagnosticContributionSet.
- A DiagnosticCustomServiceClass with a given value of attribute customServiceId shall only appear once in the context created by a DiagnosticContributionSet.
- If the subclass of DiagnosticServiceClass (except DiagnosticCustomServiceClass) appears multiple times in the context created by a DiagnosticContributionSet, then all instances of the sub-class of DiagnosticServiceClass shall have identical values for all of their attributes.

  In the case of aggregations, the number of aggregated elements shall be identical and the values of primitive attributes of aggregated elements shall again be identical.

⌋

This means that multiple instances of DiagnosticCustomServiceClass may exist in the context of a DiagnosticContributionSet, as long as the value of attribute customServiceId is different for all occurrences of DiagnosticCustomServiceClass.

The background of [constr_1329] is obviously related to the semantics of DiagnosticServiceClass, which is to define model attributes that are shared among all DiagnosticServiceInstances.

This would not be possible if more than one DiagnosticServiceClass with a diverging set of attribute values exists.

Class	***DiagnosticServiceClass*** (abstract)			
Note	This meta-class provides the ability to define common properties that are shared among all instances of sub-classes of DiagnosticServiceInstance.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticAuthenticationClass, DiagnosticClearDiagnosticInformationClass, DiagnosticClearReset EmissionRelatedInfoClass, DiagnosticComControlClass, DiagnosticControlDTCSettingClass, Diagnostic CustomServiceClass, DiagnosticDataTransferClass, DiagnosticDynamicallyDefineDataIdentifierClass, DiagnosticEcuResetClass, DiagnosticIoControlClass, DiagnosticReadDTCInformationClass, Diagnostic ReadDataByIdentifierClass, DiagnosticReadDataByPeriodicIDClass, DiagnosticReadMemoryByAddress Class, DiagnosticReadScalingDataByIdentifierClass, DiagnosticRequestControlOfOnBoardDeviceClass, DiagnosticRequestCurrentPowertrainDataClass, DiagnosticRequestDownloadClass, DiagnosticRequest EmissionRelatedDTCClass, DiagnosticRequestEmissionRelatedDTCPermanentStatusClass, Diagnostic RequestFileTransferClass, DiagnosticRequestOnBoardMonitoringTestResultsClass, DiagnosticRequest PowertrainFreezeFrameDataClass, DiagnosticRequestUploadClass, DiagnosticRequestVehicleInfo Class, DiagnosticResponseOnEventClass, DiagnosticRoutineControlClass, DiagnosticSecurityAccess Class, DiagnosticSessionControlClass, DiagnosticTransferExitClass, DiagnosticWriteDataByIdentifier Class, DiagnosticWriteMemoryByAddressClass			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.25: DiagnosticServiceClass

Class	***DiagnosticServiceInstance*** (abstract)			
Note	This represents a concrete instance of a diagnostic service.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Subclasses	*DiagnosticAuthentication*, DiagnosticClearDiagnosticInformation, DiagnosticClearResetEmissionRelated Info, DiagnosticComControl, DiagnosticControlDTCSetting, DiagnosticCustomServiceInstance, *DiagnosticDataByIdentifier*, DiagnosticDynamicallyDefineDataIdentifier, DiagnosticEcuReset, Diagnostic IOControl, *DiagnosticMemoryByAddress*, DiagnosticReadDTCInformation, DiagnosticReadDataBy PeriodicID, DiagnosticRequestControlOfOnBoardDevice, DiagnosticRequestCurrentPowertrainData, DiagnosticRequestEmissionRelatedDTC, DiagnosticRequestEmissionRelatedDTCPermanentStatus, DiagnosticRequestFileTransfer, DiagnosticRequestOnBoardMonitoringTestResults, DiagnosticRequest PowertrainFreezeFrameData, DiagnosticRequestVehicleInfo, DiagnosticResponseOnEvent, Diagnostic RoutineControl, DiagnosticSecurityAccess, DiagnosticSessionControl			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
access Permission	DiagnosticAccess Permission	0..1	ref	This represents the collection of DiagnosticAccess Permissions that allow for the execution of the referencing DiagnosticServiceInstance.. **Tags:** atp.Splitkey=accessPermission
serviceClass	DiagnosticServiceClass	0..1	ref	This represents the corresponding "class", i.e. this meta-class provides properties that are shared among all instances of applicable sub-classes of DiagnosticService Instance. The subclasses that affected by this pattern implement references to the applicable "class"-role that substantiate this abstract reference. **Stereotypes:** atpAbstract

Table 4.26: DiagnosticServiceInstance

[constr_1798] Existence of `DiagnosticServiceInstance.serviceClass`

Imposition time: CP: `IT_DiagDes`, AP: `IT_DiagDes`

⌈For each subclass of `DiagnosticServiceInstance`, a reference with the abstract role `serviceClass` shall exist to a matching subclass of `DiagnosticServiceClass`.

This rule applies unless a rule for a specific combination of matching sub-classes of `DiagnosticServiceInstance` and `DiagnosticServiceClass` exists.⌋

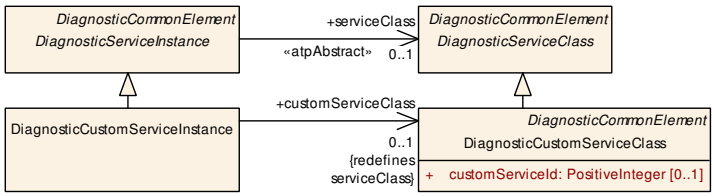

Figure 4.11: Modeling of custom service instance

[TPS_DEXT_01147] Support for custom service instance ⌈Meta-class `DiagnosticCustomServiceInstance` can be used to define the existence of an instance of a custom diagnostic service.⌋

Custom services can obviously not be configured using standardized attributes, but there is the ability to use `Sdg` in the context of `adminData` for this purpose.

There is no obligation for a given tool to be able to properly process the definition of the custom service instance.

Class	**DiagnosticCustomServiceInstance**			
Note	This meta-class has the ability to define an instance of a custom diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticCustomInstances			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
customService Class	DiagnosticCustom ServiceClass	0..1	ref	Reference to the corresponding DiagnosticCustom ServiceClass. **Tags:** atp.Splitkey=customServiceClass

Table 4.27: DiagnosticCustomServiceInstance

Class	**DiagnosticCustomServiceClass**
Note	This represents the ability to define a custom diagnostic service class and assign an ID to it. Further configuration is not foreseen from the point of view of the diagnostic extract and consequently needs to be done on the level of ECUC. **Tags:** atp.recommendedPackage=DiagnosticCustomServiceClasses
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

▽

△

Class	**DiagnosticCustomServiceClass**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
customServiceId	PositiveInteger	0..1	attr	This attribute may only be used for the definition of custom services. The values shall not overlap with existing standardized service IDs.

Table 4.28: DiagnosticCustomServiceClass

[constr_1330] Custom service identifier shall not overlap with standardized service identifiers

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈The value of the attribute DiagnosticCustomServiceClass.customServiceId shall not be set to any of the values reserved for standardized service identifiers as defined by [17, ISO 14229-1].⌋

4.3.3 Diagnostic Access Permission

This chapter discusses the concept of an *access permission* used in the context of the Dcm.

[TPS_DEXT_01139] Semantics of the references from DiagnosticAccessPermission ⌈The semantics of the references from DiagnosticAccessPermission to

- DiagnosticSession in the role diagnosticSession.
- DiagnosticSecurityLevel in the role securityLevel
- DiagnosticEnvironmentalCondition in the role environmentalCondition
- DiagnosticAuthRoleProxy in the role authenticationEnabled.

in terms of how access permission is granted is subject to the specification of the Dcm, see [10, SWS Diagnostic Communication Manager].⌋

The semantics of the model elements referenced from meta-class DiagnosticAccessPermission is described in the following sections of this document:

- DiagnosticSession is described in Chapter 4.3.4.
- DiagnosticSecurityLevel is described in Chapter 4.3.5.
- DiagnosticAuthRoleProxy is described in Chapter 4.3.6.
- DiagnosticEnvironmentalCondition is described in Chapter 4.3.7.

The DiagnosticAccessPermission is used to describe the ability (or the lack thereof) to execute a diagnostic service depending on the referenced DiagnosticSecurityLevel and DiagnosticSession (see Figure 4.12).

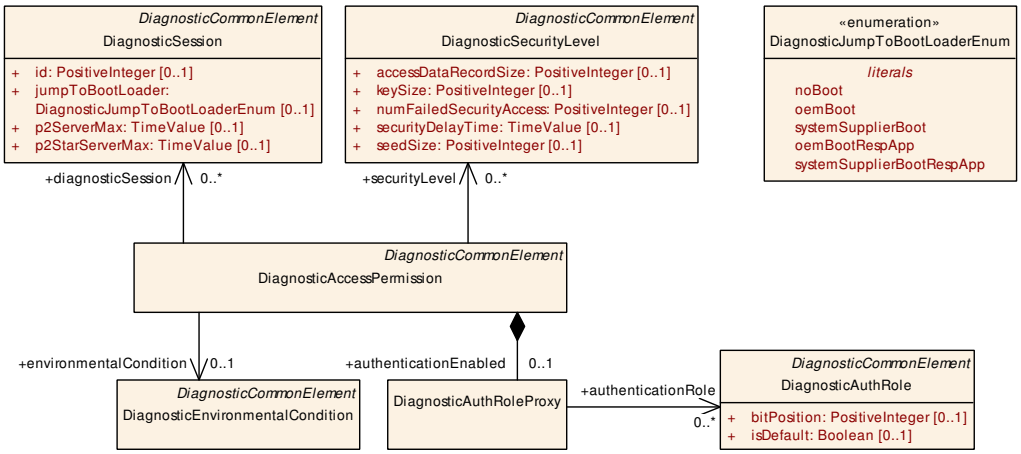

Figure 4.12: Common model elements relevant for the Dcm

[TPS_DEXT_01070] Description of textually semi-formal formulated pre- and run-conditions for the validity of `DiagnosticAccessPermission` ⌈AUTOSAR supports the description of textually formulated semi-formal pre- and run-conditions for the validity of `DiagnosticAccessPermission`.

This can be done by means of the attribute `DiagnosticAccessPermission.introduction.structuredReq`.⌋

For more details regarding the modeling of the semi-formal text please refer to Figure 4.4.

An example of how the definition of pre- and run-conditions may look like in `ARXML` is sketched in Listing B.9.

To make this approach work it is important to standardize possible values of the attribute `category` such that the semi-formal semantics of the definition of pre- and run-conditions is protected by regulation of the AUTOSAR standard.

[TPS_DEXT_01071] Standardized values of `DiagnosticAccessPermission.introduction.structuredReq` ⌈The following possible values of `DiagnosticAccessPermission.introduction.structuredReq` are standardized by AUTOSAR:

- **DIAG_ACCESS_PERM_PRE_COND**: this value describes the pre-condition of the corresponding `DiagnosticAccessPermission`.
- **DIAG_ACCESS_PERM_RUN_COND**: this value describes the run-condition of the corresponding `DiagnosticAccessPermission`.

⌋

Please find an example for the application of standardized values according to [TPS_DEXT_01071] in Chapter B.5.

Class	**DiagnosticAccessPermission**			
Note	This represents the specification of whether a given service can be accessed according to the existence of meta-classes referenced by a particular DiagnosticAccessPermission. In other words, this meta-class acts as a mapping element between several (otherwise unrelated) pieces of information that are put into context for the purpose of checking for access rights. **Tags:** atp.recommendedPackage=DiagnosticAccessPermissions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
authentication Enabled	DiagnosticAuthRole Proxy	0..1	aggr	The existence of this aggregation indicates that an authentication is foreseen. The details are clarified by the aggregated class. **Tags:** atp.Splitkey=authenticationEnabled
diagnostic Session	DiagnosticSession	*	ref	This represents the associated DiagnosticSessions **Tags:** atp.Splitkey=diagnosticSession
environmental Condition	Diagnostic EnvironmentalCondition	0..1	ref	This represents the environmental conditions associated with the access permission. **Tags:** atp.Splitkey=environmentalCondition
securityLevel	DiagnosticSecurityLevel	*	ref	This represents the associated DiagnosticSecurityLevels **Tags:** atp.Splitkey=securityLevel

Table 4.29: DiagnosticAccessPermission

Enumeration	**DiagnosticJumpToBootLoaderEnum**
Note	This enumeration contains the options for jumping to a boot loader.
Aggregated by	DiagnosticSession.jumpToBootLoader
Literal	***Description***
noBoot	This diagnostic session doesn't allow to jump to Bootloader. **Tags:** atp.EnumerationLiteralIndex=0
oemBoot	This diagnostic session allows to jump to OEM Bootloader. In this case the bootloader send the final response. **Tags:** atp.EnumerationLiteralIndex=1
oemBootRespApp	This diagnostic session allows to jump to OEM Bootloader and application sends final response. **Tags:** atp.EnumerationLiteralIndex=3
systemSupplierBoot	This diagnostic session allows to jump to System Supplier Bootloader. In this case the bootloader send the final response. **Tags:** atp.EnumerationLiteralIndex=2
systemSupplierBoot RespApp	This diagnostic session allows to jump to System Supplier Bootloader and application sends final response. **Tags:** atp.EnumerationLiteralIndex=4

Table 4.30: DiagnosticJumpToBootLoaderEnum

4.3.4 Diagnostic Session

At runtime, `DiagnosticSession`s are used to create a context for the execution of diagnostic functionality. Servers usually support a variety of different `DiagnosticSession`s. It is possible to switch between `DiagnosticSession`s at runtime.

[TPS_DEXT_01011] Semantics of `DiagnosticSession.id` ⌈The value of the attribute `DiagnosticSession.id` has a given semantics according to [17, ISO 14229-1]. For the sake of completeness, the dedicated values of `DiagnosticSession.id` are:

0x01 This represents the *default session*. This session has a specific semantics in the context of diagnostics communication such that e.g. any diagnostic service that is supposed to execute in the *default session* cannot require any reference to a `DiagnosticSecurityLevel`.

0x02 This represents the *programming session*.

0x03 This represents the *extended diagnostic session*.

0x04 This represents the *safety system diagnostic session*.

0x40..0x5F This value range is reserved for *manufacturer-specific use*.

0x60..0x7E This value range is reserved for *supplier-specific use*.

⌋

Class	**DiagnosticSession**			
Note	This meta-class represents the ability to define a diagnostic session. **Tags:** atp.recommendedPackage=DiagnosticSessions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
id	PositiveInteger	0..1	attr	This is the numerical identifier used to identify the DiagnosticSession in the scope of diagnostic workflow
jumpToBootLoader	DiagnosticJumpToBootLoaderEnum	0..1	attr	This attribute represents the ability to define whether this diagnostic session allows to jump to Bootloader (OEM Bootloader or System Supplier Bootloader). If this diagnostic session doesn't allow to jump to Bootloader the value JumpToBootLoaderEnum.noBoot shall be chosen. This Attribute is only used by the AUTOSAR Classic Platform.
p2ServerMax	TimeValue	0..1	attr	This is the session value for P2ServerMax in seconds (per Session Control). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds.
p2StarServerMax	TimeValue	0..1	attr	This is the session value for P2*ServerMax in seconds (per Session Control). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds.

Table 4.31: DiagnosticSession

4.3.5 Diagnostic Security Level

[TPS_DEXT_01012] Rationale for the modeling of the multiplicity of reference `DiagnosticAccessPermission.securityLevel` ⌈The multiplicity of the role `DiagnosticAccessPermission.securityLevel` has been set to 0..* with the following motivation:

- The `DiagnosticSession` where the attribute `DiagnosticSession.id` is set to 0x01 shall **not** be associated with a `DiagnosticSecurityLevel`.
- There are no associated `DiagnosticSecurityLevel`s required. As a consequence, the execution of the `DiagnosticServiceInstance` that references the given `DiagnosticAccessPermission` is always possible.

⌋

Class	**DiagnosticSecurityLevel**			
Note	This meta-class represents the ability to define a security level considered for diagnostic purposes. **Tags:** atp.recommendedPackage=DiagnosticSecurityLevels			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
accessDataRecordSize	PositiveInteger	0..1	attr	This represents the size of the AccessDataRecord used in GetSeed. Unit:byte.
keySize	PositiveInteger	0..1	attr	This represents the size of the security key. Unit: byte.
numFailedSecurityAccess	PositiveInteger	0..1	attr	This represents the number of failed security accesses after which the delay time is activated.
securityDelayTime	TimeValue	0..1	attr	This represents the delay time after a failed security access. Unit: second.
seedSize	PositiveInteger	0..1	attr	This represents the size of the security seed. Unit: byte.

Table 4.32: DiagnosticSecurityLevel

[constr_1419] Value of `DiagnosticSecurityLevel.accessDataRecordSize`

Imposition time: CP: `IT_DiagDes`, AP: `IT_DiagDes`

⌈If the attribute `DiagnosticSecurityLevel.accessDataRecordSize` exists then its value shall be greater than zero.⌋

4.3.6 Diagnostic Authentication

[TPS_DEXT_01188] Aggregation in the role `DiagnosticAccessPermission.authenticationEnabled` does not exist ⌈If the aggregation of meta-class `DiagnosticAuthRoleProxy` in the role `DiagnosticAccessPermission.authenticationEnabled` does not exist, then no authentication checks are done and the service is processed.⌋

Class	**DiagnosticAuthRoleProxy**			
Note	This meta-class indicates that an authentication is generally foreseen. The question whether the authentication is done in general or whether it is done role-specific depends on the existence of references to DiagAuthRole.			
Base	*ARObject*			
Aggregated by	DiagnosticAccessPermission.authenticationEnabled, DiagnosticMemoryDestinationUserDefined.authenticationEnabled			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
authentication Role	DiagnosticAuthRole	*	ref	This reference identifies the authenticationRole applicable for the enclosing DiagnosticAccessPermission. **Tags:** atp.Splitkey=authenticationRole

Table 4.33: DiagnosticAuthRoleProxy

[TPS_DEXT_01189] Semantics of attribute `DiagnosticAccessPermission.authenticationEnabled` ⌈The existence of the aggregation of meta-class `DiagnosticAuthRoleProxy` in the role `DiagnosticAccessPermission.authenticationEnabled` indicates that an authentication by means of further checks is foreseen in the context of the enclosing `DiagnosticAccessPermission`.⌋

The concrete nature of the further check mentioned in [TPS_DEXT_01189] is defined by [TPS_DEXT_01190] and [TPS_DEXT_01191].

There are use cases where diagnostic services shall explicitly not have any authentication role and the corresponding service is only accessible via the dynamic access list of certificates.

[TPS_DEXT_01190] Role `DiagnosticAccessPermission.authenticationEnabled` exists, but `DiagnosticAuthRoleProxy.authenticationRole` does not exist ⌈If no further `DiagnosticAuthRoleProxy.authenticationRole` is defined, then the service is checked against the current dynamic access list.⌋

There are use cases where diagnostic service shall be subject to specific authentication roles.

[TPS_DEXT_01191] Roles `DiagnosticAccessPermission.authenticationEnabled` and `DiagnosticAuthRoleProxy.authenticationRole` ⌈If the `DiagnosticAccessPermission`, in addition to support authentication in general, also needs to consider specific authentication roles, then these roles can be specified by means of the reference to `DiagnosticAuthRole` in the role `DiagnosticAuthRoleProxy.authenticationRole`.⌋

[TPS_DEXT_01154] Semantics of meta-class `DiagnosticAuthRole` ⌈Meta-class `DiagnosticAuthRole` provides support for role-based authentication authentication.

In particular, it supports the existence of different roles and the assignment of different rights (e.g. access a given `DiagnosticDataIdentifier`) to any of these roles.

The rights are defined in the form of a bitfield that is associated with a given role. In other words, the implementation of this “roles and rights” concept is that a bitfield is associated with a textual label (that describes the role).⌋

Please note that the value of DiagnosticAuthRole.bitPosition contributes to the "normative" definition of the "roles and rights" matrix. This matrix is typically defined individually by each OEM and applies for all projects in the OEM's organization.

Class	**DiagnosticAuthRole**			
Note	This meta-class represents the ability to specify an authentication role that can be used to deliver fine-grained access rights. **Tags:** atp.recommendedPackage=DiagnosticAuthRoles			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bitPosition	PositiveInteger	0..1	attr	This attribute allows for the specification of the position of the enclosing role in a bitfield of roles.
isDefault	Boolean	0..1	attr	This attribute indicates whether the enclosing role is considered a default role.

Table 4.34: DiagnosticAuthRole

The model relations of DiagnosticAuthRole are depicted in Figure 4.12.

Example 4.3

An example of such a matrix is sketched in Figure 4.13. For more information please refer to the document [10, SWS Diagnostic Communication Manager].

	Development	Prduction	After Sales	Extended User	ePTI	Unused	Unused	Unused
Bit	7	6	5	4	3	2	1	0
Certificate	0	0	1	1	0	0	0	0
ServiceList								
Service 0x28	0	0	1	1	0	0	0	0
Service 0x11 01	0	0	1	0	0	0	0	0
RID 0x5678	0	1	0	0	0	0	0	0

Figure 4.13: Example "roles and rights" matrix

[constr_10038] Restriction for the usage of DiagnosticAccessPermission.authenticationEnabled

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈Attribute DiagnosticAccessPermission.authenticationEnabled shall not exist if the DiagnosticAccessPermission is referenced from

- DiagnosticRequestCurrentPowertrainData
- DiagnosticRequestPowertrainFreezeFrameData

- DiagnosticRequestEmissionRelatedDTC
- DiagnosticClearResetEmissionRelatedInfo
- DiagnosticRequestOnBoardMonitoringTestResults
- DiagnosticRequestControlOfOnBoardDevice
- DiagnosticRequestVehicleInfo
- DiagnosticRequestEmissionRelatedDTCPermanentStatus
- sub-classes of DiagnosticAuthentication

⌋

4.3.7 Diagnostic Environmental Condition

In some cases, diagnostic functionality can only be executed if the vehicle is in a (safe) state that allows for the respective diagnostics function. For example, one such condition is that the vehicle is not driving, i.e. vehicle speed == 0.

The meta-class DiagnosticEnvironmentalCondition formalizes the idea of a condition which is evaluated during runtime of the ECU by looking at “environmental” states (such as the mentioned vehicle speed).

DiagnosticEnvironmentalConditions are based on the active modes in the application software or basic software or by comparison of DiagnosticDataElements with constant values. Atomic conditions can be combined by logical operations to form more complex conditions.

Please note that it is possible to create a nested hierarchy (of arbitrary depth) of DiagnosticEnvConditionFormula. This modeling is supported by the fact that DiagnosticEnvConditionFormula inherits from and, at the same time, aggregates DiagnosticEnvConditionFormulaPart.

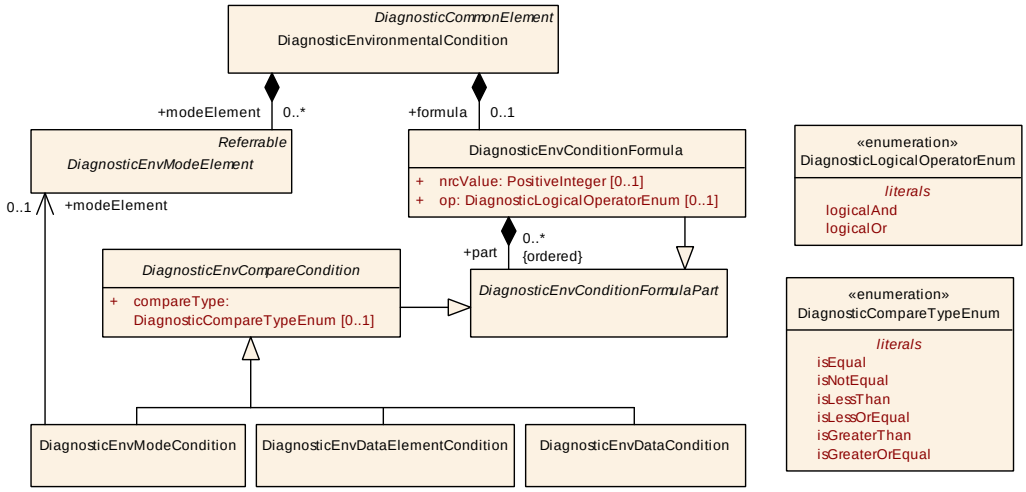

Figure 4.14: Formal modeling of the consideration of environmental conditions

Class	DiagnosticEnvironmentalCondition			
Note	The meta-class DiagnosticEnvironmentalCondition formalizes the idea of a condition which is evaluated during runtime of the ECU by looking at "environmental" states (e.g. one such condition is that the vehicle is not driving, i.e. vehicle speed == 0). **Tags:** atp.recommendedPackage=DiagnosticEnvironmentalConditions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
formula	DiagnosticEnvConditionFormula	0..1	aggr	This attribute represents the formula part of the DiagnosticEnvironmentalCondition. **Tags:** atp.Splitkey=formula
modeElement	DiagnosticEnvModeElement	*	aggr	This aggregation contains a representation of Mode Declarations in the context of a DiagnosticEnvironmentalCondition. **Tags:** atp.Splitkey=modeElement, modeElement.shortName

Table 4.35: DiagnosticEnvironmentalCondition

[constr_1799] Existence of `DiagnosticEnvironmentalCondition.formula`

Imposition time: CP: `IT_DiagDes`, AP: `IT_DiagDes`

⌈For each `DiagnosticEnvironmentalCondition`, the aggregation of `DiagnosticEnvConditionFormula` in the role `formula` shall exist.⌋

4.3.7.1 Environmental Condition Formula

The core part of a `DiagnosticEnvironmentalCondition` is the `DiagnosticEnvConditionFormula`.

[TPS_DEXT_01113] Evaluation of a `DiagnosticEnvConditionFormula` ⌈A `DiagnosticEnvConditionFormula` embodies the computation instruction that is to be evaluated at runtime to determine if the `DiagnosticEnvironmentalCondition` is currently present (i.e. the formula is evaluated to `true`) or not (otherwise).

The `DiagnosticEnvConditionFormula` itself consists of `part`s which are combined by the logical operations represented by the attribute `op`.⌋

Class	DiagnosticEnvConditionFormula
Note	A DiagnosticEnvConditionFormula embodies the computation instruction that is to be evaluated at runtime to determine if the DiagnosticEnvironmentalCondition is currently present (i.e. the formula is evaluated to true) or not (otherwise). The formula itself consists of parts which are combined by the logical operations specified by DiagnosticEnvConditionFormula.op. If a diagnostic functionality cannot be executed because an environmental condition fails then the diagnostic stack shall send a negative response code (NRC) back to the client. The value of the NRC is directly related to the specific formula and is therefore formalized in the attribute DiagnosticEnvConditionFormula.nrcValue.

▽

△

Class	DiagnosticEnvConditionFormula			
Base	*ARObject*, *DiagnosticEnvConditionFormulaPart*			
Aggregated by	DiagnosticEnvConditionFormula.part, DiagnosticEnvironmentalCondition.formula			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
nrcValue	PositiveInteger	0..1	attr	This attribute represents the concrete NRC value that shall be returned if the condition fails.
op	DiagnosticLogical OperatorEnum	0..1	attr	This attribute represents the concrete operator (supported operators: and, or) of the condition formula.
part (ordered)	DiagnosticEnvCondition FormulaPart	*	aggr	This aggregation represents the collection of formula parts that can be combined by logical operators.

Table 4.36: DiagnosticEnvConditionFormula

[constr_1800] Existence of `DiagnosticEnvConditionFormula.op`

Imposition time: CP: `IT_DiagDes`, AP: `IT_DiagDes`

⌈For each `DiagnosticEnvConditionFormula`, the attribute `op` shall exist.⌋

Enumeration	DiagnosticLogicalOperatorEnum
Note	Logical AND and OR operation (&&, ||)
Aggregated by	DiagnosticEnvConditionFormula.op
Literal	**Description**
logicalAnd	Logical AND **Tags:** atp.EnumerationLiteralIndex=0
logicalOr	Logical OR **Tags:** atp.EnumerationLiteralIndex=1

Table 4.37: DiagnosticLogicalOperatorEnum

Class	*DiagnosticEnvConditionFormulaPart* (abstract)			
Note	A DiagnosticEnvConditionFormulaPart can either be a atomic condition, e.g. a DiagnosticEnvCompare Condition, or a DiagnosticEnvConditionFormula, again, which allows arbitrary nesting.			
Base	*ARObject*			
Subclasses	*DiagnosticEnvCompareCondition*, DiagnosticEnvConditionFormula			
Aggregated by	DiagnosticEnvConditionFormula.part			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.38: DiagnosticEnvConditionFormulaPart

[TPS_DEXT_01114] `DiagnosticEnvConditionFormula` that has no `part`s ⌈A `DiagnosticEnvConditionFormula` that has no `part`s shall be evaluated to false. This rule shall apply independently of the value of `DiagnosticEnvCondition-Formula.op`.⌋

[TPS_DEXT_01115] `DiagnosticEnvConditionFormula` that has one `part` ⌈A `DiagnosticEnvConditionFormula` that has one `part` shall be evaluated to the evaluation result of this part. This rule shall apply independently of the value of `DiagnosticEnvConditionFormula.op`.⌋

[TPS_DEXT_01116] `DiagnosticEnvConditionFormula` that has more than one `part` ⌈The evaluation result of a `DiagnosticEnvConditionFormula` with more than one `part` shall be calculated by combining the results of the `part`s with the logical Operation specified by `DiagnosticEnvConditionFormula.op`.

The evaluation shall be done in a "short-cut" manner, i.e. following the rules for the evaluation of the logical operators in C.

The consequences of this approach are:

- The `part`s shall be evaluated in the specified order starting at the first element. This is why the aggregation of `part` is decorated by the `ordered` qualifier.
- after the evaluation of each `part` a check shall be executed if it is still possible that the evaluation of the remaining `part`s (to `true` or `false`) could change the overall result.
- As soon as a change of the overall result is no longer possible, i.e. any of the following conditions evaluates to `false`:
  - there are no `part`s left
  - the current `part` was evaluated to `false` and `DiagnosticEnvConditionFormula.op == DiagnosticLogicalOperatorEnum.logicalAnd`
  - the current `part` was evaluated to `true` and `DiagnosticEnvConditionFormula.op == DiagnosticLogicalOperatorEnum.logicalOr`

  the evaluation of the `part`s shall be finalized and the evaluation result of the current `part` shall be considered the overall evaluation result of the formula.

  Regarding the strategy for returning `NRC` values please refer to the specification of the [10, SWS Diagnostic Communication Manager].

⌋

[TPS_DEXT_01117] Semantics of `DiagnosticEnvConditionFormula.nrcValue` ⌈If a diagnostic functionality cannot be executed because an environmental condition fails, i.e. the formula is evaluated to false, then the diagnostic stack shall send an optional negative response code (`NRC`) back to the client (if it is present).

If no value for `NRC` is defined then a pre-defined `NRC` value as defined in the [10, SWS Diagnostic Communication Manager] will be sent back. The value of the `NRC` is directly related to the specific `DiagnosticEnvConditionFormula` and is therefore formalized in the attribute `DiagnosticEnvConditionFormula.nrcValue`.⌋

[constr_1464] Allowed value range of DiagnosticEnvConditionFormula.nrcValue

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈The value of attribute DiagnosticEnvConditionFormula.nrcValue shall be limited to the interval [1..255].⌋

The rationale for the existence of [constr_1464] is provided by [17, ISO 14229-1].

4.3.7.2 Atomic Conditions

Atomic conditions in the context of a DiagnosticEnvConditionFormula are described by means of meta-class DiagnosticEnvCompareConditions. The formulation of DiagnosticEnvCompareConditions is based on the idea of a comparison at runtime of some variable data with a constant value.

The actual type of the comparison (==, !=, <, <=, ...) is specified by means of the attribute DiagnosticEnvCompareCondition.compareType.

Class	***DiagnosticEnvCompareCondition*** (abstract)			
Note	DiagnosticCompareConditions are atomic conditions. They are based on the idea of a comparison at runtime of some variable data with something constant. The type of the comparison (==, !=, <, <=, ...) is specified in DiagnosticCompareCondition.compareType.			
Base	*ARObject*, *DiagnosticEnvConditionFormulaPart*			
Subclasses	DiagnosticEnvDataCondition, DiagnosticEnvDataElementCondition, DiagnosticEnvModeCondition			
Aggregated by	DiagnosticEnvConditionFormula.part			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
compareType	DiagnosticCompare TypeEnum	0..1	attr	This attributes represents the concrete type of the comparison.

Table 4.39: DiagnosticEnvCompareCondition

[constr_1801] Existence of DiagnosticEnvCompareCondition.compareType

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticEnvCompareCondition, that attribute compareType shall exist.⌋

DiagnosticEnvCompareCondition is an abstract meta-class that acts as a base class for two concrete meta-classes designed to handle different operand types.

Enumeration	**DiagnosticCompareTypeEnum**
Note	Enumeration for the type of a comparison of values usually expressed by the following operators: ==, !=, <, <=, >, >=
Aggregated by	*DiagnosticEnvCompareCondition*.compareType
Literal	***Description***

▽

△

Enumeration	DiagnosticCompareTypeEnum
isEqual	equal **Tags:** atp.EnumerationLiteralIndex=0
isGreaterOrEqual	greater than or equal **Tags:** atp.EnumerationLiteralIndex=5
isGreaterThan	greater than **Tags:** atp.EnumerationLiteralIndex=4
isLessOrEqual	less than or equal **Tags:** atp.EnumerationLiteralIndex=3
isLessThan	less than **Tags:** atp.EnumerationLiteralIndex=2
isNotEqual	not equal **Tags:** atp.EnumerationLiteralIndex=1

Table 4.40: DiagnosticCompareTypeEnum

The specific sub-classes (the details are explained in Chapter 4.3.7.2.1, Chapter 4.3.7.2.2, and Chapter 4.3.7.2.3) of `DiagnosticEnvCompareCondition` support a different set of attributes of `DiagnosticCompareTypeEnum` for setting the value of attribute `DiagnosticEnvCompareCondition.compareType`.

4.3.7.2.1 Data Condition to DID

[TPS_DEXT_01118] Semantics of `DiagnosticEnvDataCondition` ⌈The meta-class `DiagnosticEnvDataCondition` represents an atomic condition that compares the current value of the referenced `DiagnosticDataElement` with a constant value defined by the `ValueSpecification` aggregated in the role `compareValue`.⌋

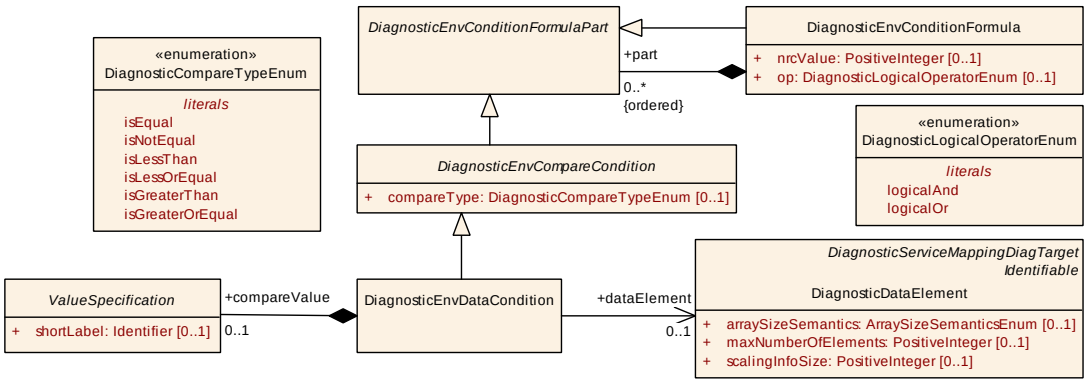

Figure 4.15: Usage of `DiagnosticEnvDataCondition`

[TPS_DEXT_01168] Allowed values of `compareType` in the context of a `DiagnosticEnvDataCondition` ⌈Within the context of a `DiagnosticEnvDataCondition` **all values** of `DiagnosticCompareTypeEnum` are supported for the inherited attribute `compareType`.⌋

Class	**DiagnosticEnvDataCondition**			
Note	A DiagnosticEnvDataCondition is an atomic condition that compares the current value of the referenced DiagnosticDataElement with a constant value defined by the ValueSpecification. All compareTypes are supported.			
Base	*ARObject*, *DiagnosticEnvCompareCondition*, *DiagnosticEnvConditionFormulaPart*			
Aggregated by	DiagnosticEnvConditionFormula.part			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
compareValue	ValueSpecification	0..1	aggr	This attribute represents a fixed compare value taken to evaluate the compare condition.
dataElement	DiagnosticDataElement	0..1	ref	This reference represents the related diagnostic data element.

Table 4.41: DiagnosticEnvDataCondition

[constr_1802] Existence of DiagnosticEnvDataCondition.compareValue

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticEnvDataCondition, that attribute compareValue shall exist.⌋

[constr_1803] Existence of DiagnosticEnvDataCondition.dataElement

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticEnvDataCondition, that attribute dataElement shall exist.⌋

4.3.7.2.2 Data Condition to Data Element owned by Application

[TPS_DEXT_01165] Semantics of DiagnosticEnvDataElementCondition ⌈The meta-class DiagnosticEnvDataElementCondition represents an atomic condition that compares the current value of the referenced DataPrototype with a constant value defined by the ValueSpecification aggregated in the role compareValue.⌋

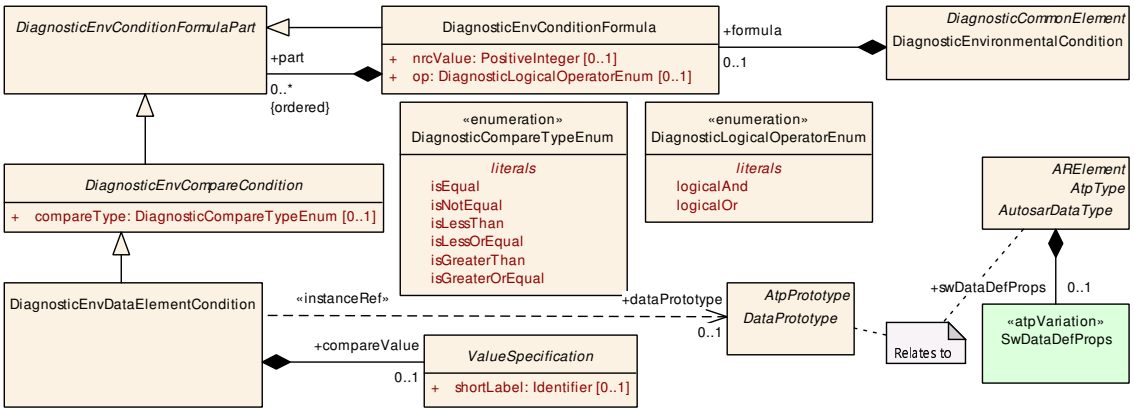

Figure 4.16: Usage of DiagnosticEnvDataElementCondition in a top-down development scenario

[TPS_DEXT_01166] Usage of DiagnosticEnvDataElementCondition in top-down and bottom-up scenarios ⌈Meta-class DiagnosticEnvDataElement-Condition supports the definition of access to a DataPrototype owned by the application:

- If the application software exists (i.e. a top-down development process is in place) at the time when the DiagnosticEnvDataElementCondition is created, then the reference in the role dataPrototype can be used to identify the target DataPrototype right away.
- If the application software does **not** exist (i.e. a bottom-up development process is in place) at the time when the DiagnosticEnvDataElementCondition is created, then the aggregation of SwDataDefProps in the role swDataDef-Props shall exist.

  In this case, the properties (baseTypeEncoding, baseTypeSize) of the DataPrototype (most likely identified by the integrator) taken for the comparison can be formally described.

⌋

Class	**DiagnosticEnvDataElementCondition**			
Note	This meta-class represents the ability to formulate a diagnostic environment condition based on the value of a data element owned by the application software.			
Base	*ARObject*, *DiagnosticEnvCompareCondition*, *DiagnosticEnvConditionFormulaPart*			
Aggregated by	DiagnosticEnvConditionFormula.part			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
compareValue	ValueSpecification	0..1	aggr	This aggregation represents the definition of the compare value against which the value taken from the application software shall be compared.
dataPrototype	DataPrototype	0..1	iref	This instanceRef represent the ability to access a data element owned by the application software on the AUTOSAR classic platform. **InstanceRef implemented by:** DataPrototypeInSystem InstanceRef
swDataDef Props	SwDataDefProps	0..1	aggr	Via this aggregation it is possible to describe the properties of the data that is obtained from the application for the environmental condition.

Table 4.42: DiagnosticEnvDataElementCondition

[constr_10115] Existence of attributes of meta-class DiagnosticEnvData-ElementCondition if the reference in the role dataPrototype exists

Imposition time: CP: IT_DiagDes

⌈If the reference in the role DiagnosticEnvDataElementCondition.data-Prototype exists, then

- the aggregation in the role compareValue shall exist and
- the aggregation in the role swDataDefProps shall not exist.

⌋

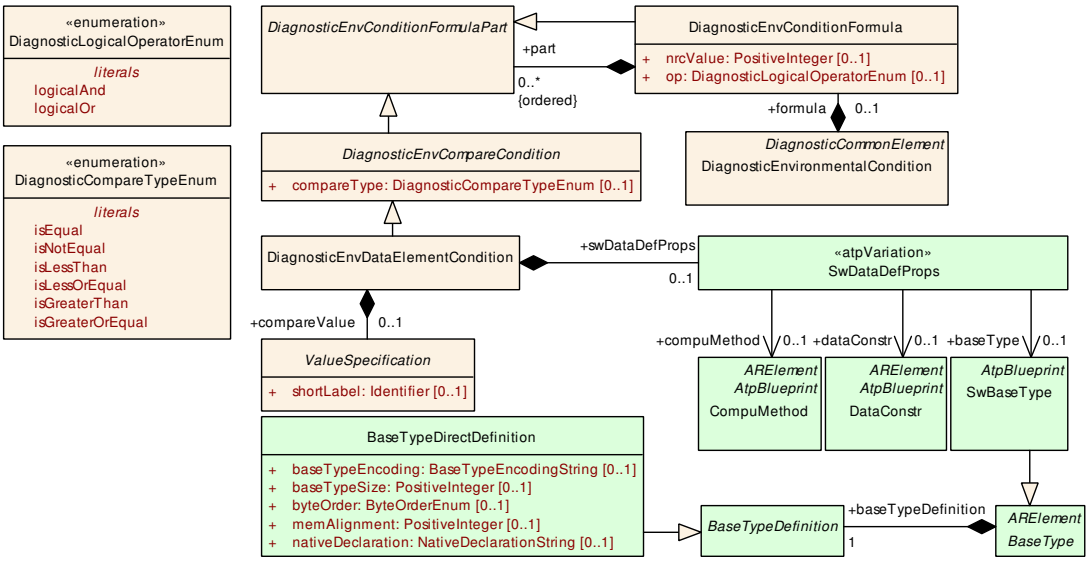

Figure 4.17: Usage of `DiagnosticEnvDataElementCondition` in a bottom-up development scenario

[constr_10116] Existence of attributes of meta-class `DiagnosticEnvDataElementCondition` if the reference in the role `dataPrototype` does not exist

Imposition time: CP: `IT_DiagDes`

⌈If the reference in the role `DiagnosticEnvDataElementCondition.dataPrototype` does **not** exist, then the aggregations in the role

- `compareValue` and
- `swDataDefProps`

shall exist.⌋

[constr_10117] Existence of attributes of `DiagnosticEnvDataElementCondition.swDataDefProps`

Imposition time: CP: `IT_DiagDes`

⌈

Attribute of `SwDataDefProps`	Attribute Existence
`baseType`	1
`compuMethod`	0..1
`dataConstr`	0..1

⌋

[TPS_DEXT_01167] Relevance of `CompuMethod` and `DataConstr` for the definition of `DiagnosticEnvDataElementCondition`

⌈The references `SwDataDefProps.compuMethod` and `dataConstr` can be used to define the data semantics of the data obtained from the application software such that – where applicable –

the compareValue can be defined in the physical domain, i.e. as a ApplicationValueSpecification.⌋

4.3.7.2.3 Mode Condition

[TPS_DEXT_01119] Semantics of DiagnosticEnvModeCondition ⌈The meta-class DiagnosticEnvModeCondition represents an atomic condition that compares the current value of the referenced ModeDeclarationGroupPrototype with the value of a ModeDeclaration taken as the reference value.⌋

The concrete modeling of this aspect is sketched in Figure F.4.

Please note that the ModeDeclarationGroupPrototype as well as the applicable ModeDeclaration are both referenced in the concrete modeling of the attribute DiagnosticEnvSwcModeElement.mode.

The idea behind this modeling approach (for more information, please refer to Figure F.4) is that the ModeDeclaration can only be a member of the specific ModeDeclarationGroup taken to type the respective ModeDeclarationGroupPrototype anyway.

Admittedly, this relation could be expressed by means of two references and a formal constraint **or** it could be expressed by way of a **single** reference that's implying an interpreting of the model in a specific way. AUTOSAR opts for the latter alternative, which is detailed by [TPS_DEXT_01120].

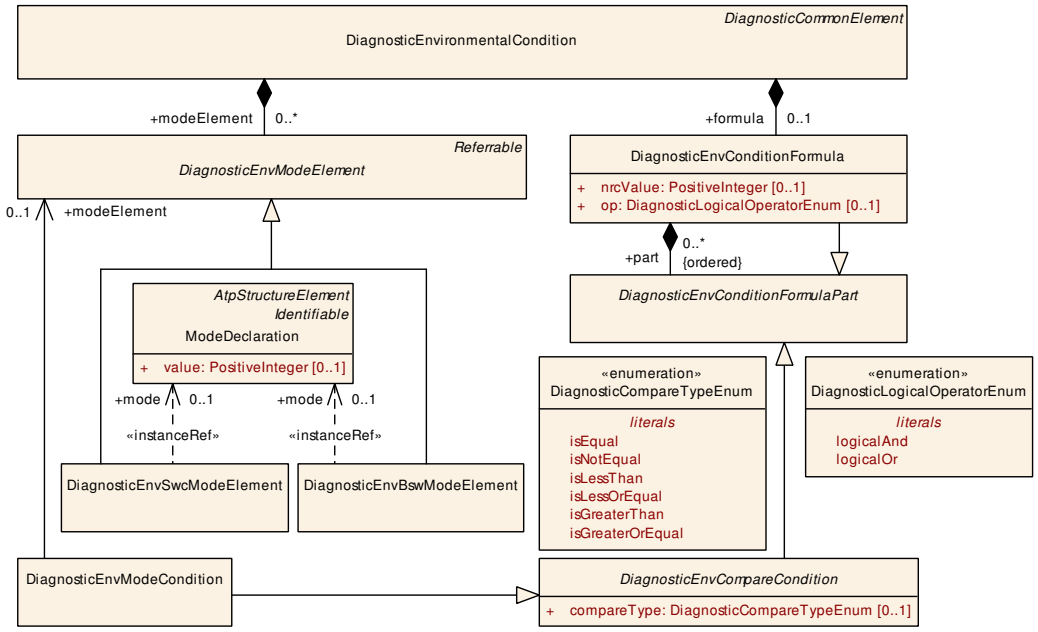

Figure 4.18: Modeling of the environmental mode condition

[TPS_DEXT_01120] Comparison of the value of a ModeDeclarationGroupPrototype with a ModeDeclaration ⌈For the comparison of the value of a ModeDeclarationGroupPrototype with a ModeDeclaration, two alternatives apply, depending on whether mode condition is executed in application software (swc) or basic software (bsw):

- The ModeDeclarationGroupPrototype referenced in the role DiagnosticEnvSwcModeElement.mode.contextModeDeclarationGroup shall be compared to the ModeDeclaration referenced in the role DiagnosticEnvSwcModeElement.mode.targetMode.
- The ModeDeclarationGroupPrototype referenced in the role DiagnosticEnvBswModeElement.mode.contextModeDeclarationGroup shall be compared to the ModeDeclaration referenced in the role DiagnosticEnvBswModeElement.mode.targetMode.

⌋

Please note that the two alternatives mentioned in [TPS_DEXT_01120] are described in Figure 4.18.

[constr_1466] Allowed values of compareType in the context of a DiagnosticEnvModeCondition

Imposition time: CP: IT_DiagDes

⌈Within the context of a DiagnosticEnvDataCondition **only a subset** of the values of DiagnosticCompareTypeEnum is supported for the inherited attribute compareType, namely:

- DiagnosticCompareTypeEnum.isEqual
- DiagnosticCompareTypeEnum.isNotEqual

⌋

[constr_1467] References in DiagnosticEnvModeCondition

Imposition time: CP: IT_DiagDes

⌈In a DiagnosticEnvModeCondition the reference modeElement shall only point to a DiagnosticEnvModeElement that is aggregated inside the same DiagnosticEnvironmentalCondition as the DiagnosticEnvModeCondition itself.⌋

Please note that the main benefit of the existence of the dedicated meta-class DiagnosticEnvModeElement is to keep the (serialized) model clean. Given the fulfillment of [constr_1467], the potentially lengthy InstanceRef for identifying the operands of a mode comparison does not (if applicable) have to be repeated but can be reused multiple times in the context of the enclosing DiagnosticEnvironmentalCondition.

Class	DiagnosticEnvModeCondition			
Note	DiagnosticEnvModeCondition are atomic condition based on the comparison of the active Mode Declaration in a ModeDeclarationGroupProtoype with the constant value of a ModeDeclaration. The formulation of this condition uses only one DiagnosticEnvElement, which contains enough information to deduce the variable part (i.e. the part that changes at runtime) as well as the constant part of the comparison. Only DiagnosticCompareTypeEnum.isEqual or DiagnosticCompareTypeEnum.isNotEqual are eligible values for DiagnosticAtomicCondition.compareType.			
Base	*ARObject*, *DiagnosticEnvCompareCondition*, *DiagnosticEnvConditionFormulaPart*			
Aggregated by	DiagnosticEnvConditionFormula.part			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
modeElement	DiagnosticEnvMode Element	0..1	ref	This reference represents both the ModeDeclaration GroupPrototype and the ModeDeclaration relevant for the mode comparison.

Table 4.43: DiagnosticEnvModeCondition

[constr_1804] Existence of DiagnosticEnvModeCondition.modeElement

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEnvModeCondition, that attribute modeElement shall exist.⌋

Class	***DiagnosticEnvModeElement*** (abstract)			
Note	All ModeDeclarations that are referenced in a DiagnosticEnvModeCondition shall be defined as a DiagnosticEnvModeElement of this DiagnosticEnvironmentalCondition. This concept keeps the ARXML clean: It avoids that the DiagnosticEnvConditionFormula is cluttered by lengthy InstanceRef definitions. Furthermore, it allows that an InstanceRef only needs to be defined once and can be used multiple times in the different DiagnosticEnvModeConditions.			
Base	*ARObject*, *Referrable*			
Subclasses	DiagnosticEnvBswModeElement, DiagnosticEnvSwcModeElement			
Aggregated by	DiagnosticEnvironmentalCondition.modeElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.44: DiagnosticEnvModeElement

Class	DiagnosticEnvSwcModeElement			
Note	This meta-class represents the ability to refer to a ModeDeclaration in a concrete System context.			
Base	*ARObject*, *DiagnosticEnvModeElement*, *Referrable*			
Aggregated by	DiagnosticEnvironmentalCondition.modeElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
mode	ModeDeclaration	0..1	iref	This reference identifies both the ModeDeclarationGroup Prototype and the ModeDeclaration for the specific mode comparison. **Tags:** atp.Splitkey=mode, mode.contextComposition, mode.contextComponent, mode.contextPPort, mode.contextModeDeclarationGroup, mode.targetMode **InstanceRef implemented by:** PModeInSystemInstance Ref

Table 4.45: DiagnosticEnvSwcModeElement

[constr_1805] Existence of `DiagnosticEnvSwcModeElement.mode`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticEnvSwcModeElement`, that attribute `mode` shall exist.⌋

Class	**DiagnosticEnvBswModeElement**			
Note	This meta-class represents the ability to refer to a specific ModeDeclaration in the scope of a BswModule Description.			
Base	*ARObject*, *DiagnosticEnvModeElement*, *Referrable*			
Aggregated by	DiagnosticEnvironmentalCondition.modeElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
mode	ModeDeclaration	0..1	iref	This reference identifies both the ModeDeclarationGroup Prototype and the ModeDeclaration for the specific mode comparison. **Tags:** atp.Splitkey=mode, mode.contextModeDeclaration Group, mode.targetMode **InstanceRef implemented by:** ModeInBswModule DescriptionInstanceRef

Table 4.46: DiagnosticEnvBswModeElement

[constr_1806] Existence of `DiagnosticEnvBswModeElement.mode`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticEnvBswModeElement`, that attribute `mode` shall exist.⌋

4.3.8 Diagnostic Services supported by AUTOSAR

The following sub-chapters describe the modeling of the collection of relevant diagnostic services as defined in the [17, ISO 14229-1]. This means that the definition of the AUTOSAR `DiagnosticExtract` does not explicitly support the total collection of diagnostic services as defined by [17, ISO 14229-1].

Some diagnostic services compiled in this document define so-called sub-functions that need to be identified to fully specify the nature of the respective diagnostic service.

[TPS_DEXT_01013] Specification of sub-functions by means of attribute `DiagnosticServiceInstance.category` ⌈

In all cases where a diagnostic service defines a sub-function according to [17, ISO 14229-1], the value of the attribute `category` of the applicable sub-class of `DiagnosticServiceInstance` can be used to specify the applicable sub-function as a textual token.

Constraints are defined to clarify the existence of standardized values of the attribute `category` for the given sub-function. This implies that an instance of the given sub-class of `DiagnosticServiceInstance` only has a single sub-function at a time.⌋

[TPS_DEXT_01045] Supported diagnostic services in the scope of this document ⌈

ID	Service
0x10	SessionControl
0x11	EcuReset
0x14	ClearDiagnosticInformation
0x19	ReadDTCInformation
0x22	ReadDataByIdentifier
0x23	ReadMemoryByAddress
0x24	ReadScalingDataByIdentifier
0x27	SecurityAccess
0x28	CommunicationControl
0x29	Authentication
0x2A	ReadDataByPeriodicIdentifier
0x2C	DynamicallyDefineDataIdentifier
0x2E	WriteDataByIdentifier
0x2F	IOControl
0x31	RoutineControl
0x34	RequestDownload
0x35	RequestUpload
0x36	TransferData
0x37	RequestTransferExit
0x3D	WriteMemoryByAddress
0x85	ControlDTCSetting
0x86	ResponseOnEvent

⌋

[TPS_DEXT_01014] Possible values of the `category` attribute for diagnostic services ⌈AUTOSAR claims the right to standardize the possible values of the attribute `category` for given diagnostic services.⌋

Example 4.4

```
<AR-PACKAGE>
  <SHORT-NAME>DiagnosticExtractExample</SHORT-NAME>
  <ELEMENTS>
    <DIAGNOSTIC-ECU-RESET>
      <SHORT-NAME>ResetTheHardWay</SHORT-NAME>
      <CATEGORY>ACME_REALLY_HARD_RESET</CATEGORY>
    </DIAGNOSTIC-ECU-RESET>
  </ELEMENTS>
</AR-PACKAGE>
```

Listing 4.1: Example for the definition of a custom `category`

If applicable, AUTOSAR allows for the usage of values of the attribute `category` other than the standardized values.

In this case, however, proprietary values of the attribute `category` shall be prefixed with a company-specific name fragment (as demonstrated in Example 4.4) in order to avoid collisions that could occur if or when the list of possible values claimed by the AUTOSAR standard itself is extended.

4.3.8.1 SessionControl

This chapter describes the modeling of diagnostic services `SessionControl` (0x10). The obvious goal of the service is to support the switching from one diagnostic session to another.

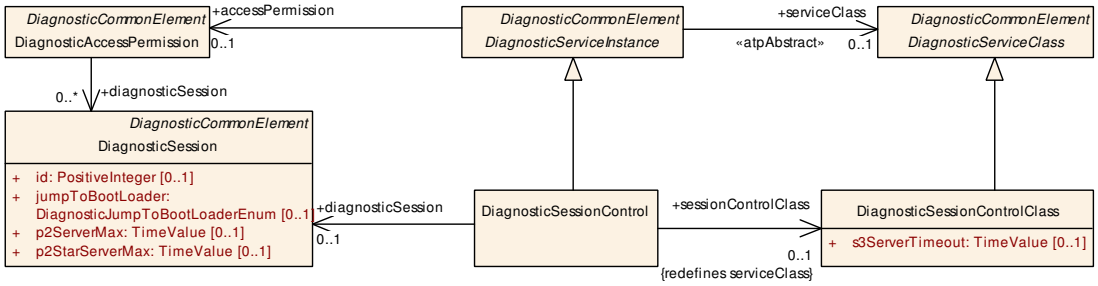

Figure 4.19: Modeling of diagnostic service `SessionControl` (0x10)

[TPS_DEXT_01081] Modeling of `DiagnosticSessionControl` ⌈For the purpose of providing a means to specify the switching from one diagnostic session to another diagnostic session, `DiagnosticSessionControl` refers to a `DiagnosticSession` in the role `diagnosticSession`.⌋

According to [17, ISO 14229-1], the diagnostic service `SessionControl` defines sub-functions.

[TPS_DEXT_01039] Identification of the sub-function of `DiagnosticSessionControl` ⌈In the case of `DiagnosticSessionControl` it would not be a good idea to encode the applicable sub-function by means of the attribute `DiagnosticSessionControl.category`.

Actually, the possible sub-functions are strongly related to the concept of the diagnostic session, represented by the meta-class `DiagnosticSession`.

The latter, in turn, has an attribute `id` that directly corresponds to the number of the applicable sub-function for `DiagnosticSessionControl`.

In other words, the sub-function of `DiagnosticSessionControl` is identified by means of the reference `DiagnosticSessionControl.diagnosticSession`.⌋

[TPS_DEXT_01082] Existence of `DiagnosticSessionControl.diagnosticSession` ⌈The idea of modeling the sub-function of `DiagnosticSessionControl` by means of the reference `DiagnosticSessionControl.diagnosticSession` implies that the applicable `DiagnosticSession` already exists at the time when the given `DiagnosticSessionControl` is created.

It is assumed that this will always be the case because the definition of DiagnosticSessions is part of laying the groundwork[5] for diagnostic communication.⌋

It is hard to foresee a scenario where the DiagnosticSessions are defined near the very end of the work-flow that leads to a complete DiagnosticExtract.

Class	DiagnosticSessionControl			
Note	This represents an instance of the "Session Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticSessionControls			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnostic Session	DiagnosticSession	0..1	ref	This represents the applicable DiagnosticSessions **Tags:** atp.Splitkey=diagnosticSession
sessionControl Class	DiagnosticSession ControlClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticSessionControl in the given context. **Tags:** atp.Splitkey=sessionControlClass

Table 4.47: DiagnosticSessionControl

[constr_1818] Existence of reference in the role DiagnosticSessionControl.diagnosticSession

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticSessionControl, the reference to DiagnosticSession in the role diagnosticSession shall exist.⌋

Class	DiagnosticSessionControlClass			
Note	This meta-class contains attributes shared by all instances of the "Session Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticSessionControls			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
s3Server Timeout	TimeValue	0..1	attr	Time for the server to keep a diagnostic session other than the default session active while not receiving any diagnostic request message.

Table 4.48: DiagnosticSessionControlClass

[5] This is similar to the definition of commonly used data types in a software development project

[constr_10440] Restriction for the minimum value of attribute DiagnosticSessionControlClass.s3ServerTimeout

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈The value of attribute DiagnosticSessionControlClass.s3ServerTimeout shall be **greater than or equal to** 5.0.⌋

4.3.8.2 SecurityAccess

This chapter describes the modeling of diagnostic services SecurityAccess (0x27).

[TPS_DEXT_01053] Existence of DiagnosticSecurityAccess.securityLevel ⌈The configuration of a given DiagnosticSecurityAccess is considered incomplete until the reference in the role DiagnosticSecurityAccess.securityLevel exists.⌋

The meaning of [TPS_DEXT_01053] is that the reference may be missing in intermediate steps of the configuration work flow. But at the point in time when ECU configuration is generated from the DiagnosticExtract the reference is needed to able to make sense of the model for the given DiagnosticSecurityAccess.

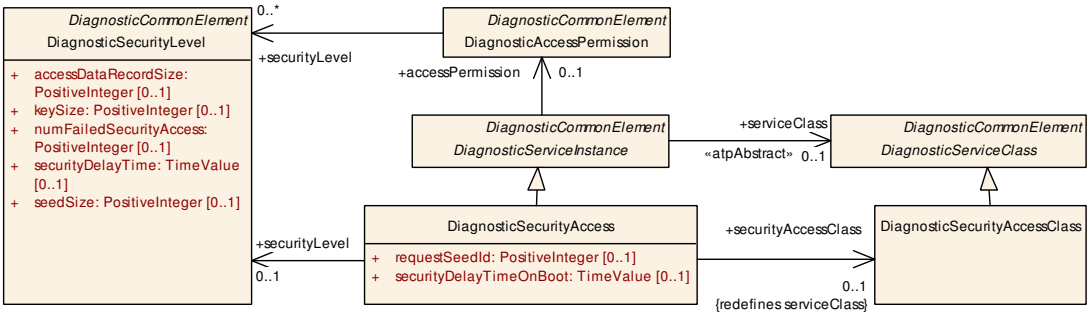

Figure 4.20: Modeling of diagnostic service SecurityAccess (0x27)

Please note that (as already explained in Chapter 4.3.8) the sub-functions of this service are modeled by means of the category attribute.

In response to conceptual differences between many other diagnostic services and SecurityAccess, the applicable sub-functions for the diagnostic service SecurityAccess are **not** defined by means of the attribute DiagnosticSecurityAccess.category.

[TPS_DEXT_01036] Work-flow within the execution of the diagnostic service SecurityAccess ⌈The work-flow within the execution of the diagnostic service SecurityAccess basically boils down to the tester sending the request to obtain a *seed* from the diagnostic stack and then sending back a *key* to the stack.

Thus, the sub-functions could be generically be described as *requestSeed* and *sendKey*, which is precisely what the [17, ISO 14229-1] does.

According to this logic, the *requestSeed* could get a specific number assigned to identify the sub-function and then the *sendKey* sub-function would get assigned the number of the *requestSeed* sub-function + 1. Again, this is fully in line with the [17, ISO 14229-1].⌋

However, there is further dimension to take into account, namely the DiagnosticSecurityLevel. According to [17, ISO 14229-1], different security levels make different numbers for the sub-function identifier.

[TPS_DEXT_01037] Semantics of DiagnosticSecurityAccess.requestSeedId ⌈The attribute DiagnosticSecurityAccess.requestSeedId shall be used to define the number of the sub-function of the diagnostic service SecurityAccess according to the intended security level.⌋

[constr_1342] Possible values for DiagnosticSecurityAccess.requestSeedId

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈The value of the attribute DiagnosticSecurityAccess.requestSeedId **shall only be set to an odd number**[6].

The supported value range consists of the following list:

- all odd numbers in the closed interval **0x01** .. **0x41**
- **0x5F** (this corresponds to the case of *end-of-life activation of on-board pyrotechnic devices according to [18, ISO 26021-2]*)
- all odd numbers in the closed interval **0x61** .. **0x7E**

⌋

In contrast to a similar situation in the case of the diagnostic service SessionControl (see Chapter 4.3.8.1), there is no real evidence that a DiagnosticSecurityLevel always exists before the referencing DiagnosticSecurityAccess is created in order to properly establish the reference in the role DiagnosticSecurityAccess.securityLevel.

[TPS_DEXT_01038] Motivation for making the reference DiagnosticSecurityAccess.securityLevel ≪atpSplitable≫ ⌈The reference DiagnosticSecurityAccess.securityLevel needs to be decorated with the stereotype ≪atpSplitable≫ in order to advertise the idea that the reference to a corresponding DiagnosticSecurityLevel is created (potentially in a different artifact) some time after the actual creation of the given DiagnosticSecurityAccess.⌋

Of course, if the DiagnosticSecurityLevel factually exists before the definition of DiagnosticSecurityAccess the reference can directly be inserted into the model.

[6] The even numbers are reserved for the identification of the corresponding *sendKey* sub-function, as explained by [TPS_DEXT_01036].

Class	**DiagnosticSecurityAccess**			
Note	This represents an instance of the "Security Access" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticSecurityAccesss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
requestSeedId	PositiveInteger	0..1	attr	This would be 0x01, 0x03, 0x05, ... The sendKey id can be computed by adding 1 to the requestSeedId
securityAccessClass	DiagnosticSecurityAccessClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticSecurityAccess in the given context. **Tags:** atp.Splitkey=securityAccessClass
securityDelayTimeOnBoot	TimeValue	0..1	attr	Start delay timer on power on in seconds. This delay indicates the time after ECU boot power-on where no security access request is accepted.
securityLevel	DiagnosticSecurityLevel	0..1	ref	This reference identifies the applicable security level for the security access. **Tags:** atp.Splitkey=securityLevel

Table 4.49: DiagnosticSecurityAccess

[constr_1816] Existence of attribute DiagnosticSecurityAccess.requestSeedId

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticSecurityAccess, the attribute requestSeedId shall exist at least once.⌋

[constr_1817] Existence of attribute DiagnosticSecurityAccess.securityLevel

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticSecurityAccess, the attribute securityLevel shall exist at least once.⌋

[constr_10045] Existence of attribute DiagnosticSecurityAccess.securityDelayTimeOnBoot

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈Attribute DiagnosticSecurityAccess.securityDelayTimeOnBoot shall exist.⌋

Class	**DiagnosticSecurityAccessClass**			
Note	This meta-class contains attributes shared by all instances of the "Security Access" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticSecurityAccesss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.50: DiagnosticSecurityAccessClass

4.3.8.3 Authentication

This chapter describes the modeling of the diagnostic service `Authentication` (0x29). The purpose of this service is to allow the tester to identify itself to the ECU, and thereby gain access to services allowed for its “role”.

The modeling of the diagnostic service Authentication follows the pattern established for other diagnostic services. In particular, meta-class `DiagnosticAuthentication` represents an instance of the service while meta-class `DiagnosticAuthenticationClass` carries configuration that is shared by all instances of the service.

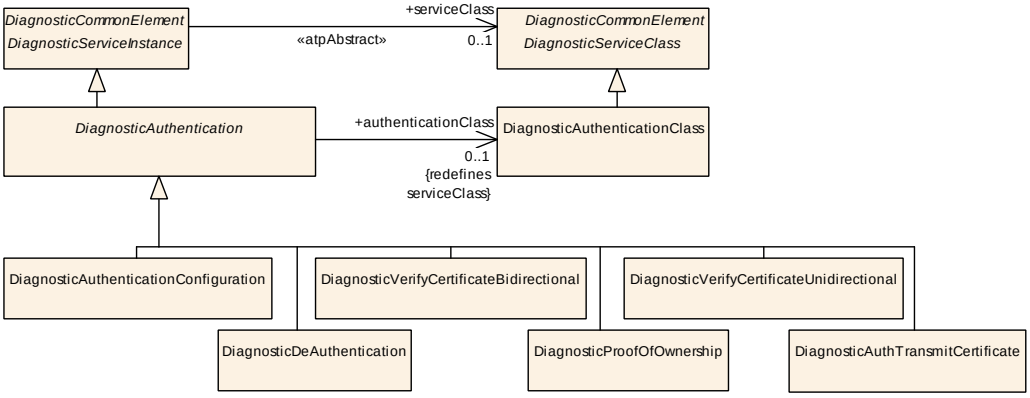

Figure 4.21: Modeling of diagnostic service `Authentication` (0x29)

[TPS_DEXT_01158] Sub-Functions for diagnostic service `Authentication` ⌈The following sub functions are supported for the diagnostic service `Authentication`:

- De-authentication, formalized by meta-class `DiagnosticDeAuthentication`.
- Verify certificate unidirectional, formalized by meta-class `DiagnosticVerifyCertificateUnidirectional`.
- Verify certificate bidirectional, formalized by meta-class `DiagnosticVerifyCertificateBidirectional`.
- Proof of ownership, formalized by meta-class `DiagnosticProofOfOwnership`.

- Authentication configuration, formalized by meta-class Diagnostic-AuthenticationConfiguration.

⌋

[TPS_DEXT_01159] Supported authentication methods for diagnostic service `Authentication` ⌈The modeling of the Authentication service only provides support for a PKI certificate exchange.⌋

[constr_10091] Mandatory subfunction of diagnostic service `Authentication`

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈If the diagnostic service Authentication is supported, then the following sub-functions shall be configured:

- De-authentication, formalized by meta-class DiagnosticDeAuthentication.
- Proof of ownership, formalized by meta-class DiagnosticProofOf-Ownership.
- Authentication configuration, formalized by meta-class Diagnostic-AuthenticationConfiguration.
- One of
  - Verify certificate unidirectional, formalized by meta-class Diagnostic-VerifyCertificateUnidirectional.
  - Verify certificate bidirectional, formalized by meta-class Diagnostic-VerifyCertificateBidirectional.

⌋

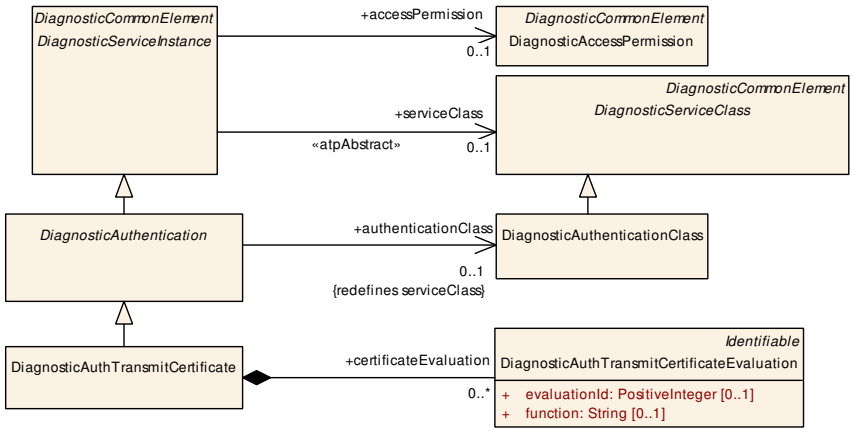

Figure 4.22: Modeling of certificate transmission within the context of the diagnostic service `Authentication` (0x29)

[TPS_DEXT_01192] Semantics of the attribute DiagnosticAuthTransmitCertificateEvaluation.evaluationId ⌈The attribute DiagnosticAuthTransmitCertificateEvaluation.evaluationId is used to define the evaluation id used for the sub-function transferCertificate (0x04).

The value of this attribute is not standardized by either the [17, ISO 14229-1] or AUTOSAR and therefore has an OEM-specific meaning.⌋

Class	***DiagnosticAuthentication*** (abstract)			
Note	This meta-class represents the ability to configure the usage of the UDS service Authentication in the Diagnostic extract.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticAuthTransmitCertificate, DiagnosticAuthenticationConfiguration, DiagnosticDeAuthentication, DiagnosticProofOfOwnership, DiagnosticVerifyCertificateBidirectional, DiagnosticVerifyCertificate Unidirectional			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
authentication Class	Diagnostic AuthenticationClass	0..1	ref	This represents the corresponding "class", i.e. this meta-class provides properties that are shared among all instances of applicable sub-classes of DiagnosticService Instance. The subclasses that affected by this pattern implement references to the applicable "class"-role that substantiate this abstract reference. **Tags:** atp.Splitkey=authenticationClass

Table 4.51: DiagnosticAuthentication

Class	**DiagnosticAuthenticationClass**			
Note	This meta-class contains configuration shared by all instances of the Authentication diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticAuthentications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.52: DiagnosticAuthenticationClass

Class	**DiagnosticAuthenticationConfiguration**			
Note	This meta-class represents the subfunction to configure the authentication. **Tags:** atp.recommendedPackage=DiagnosticAuthentications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAuthentication*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.53: DiagnosticAuthenticationConfiguration

Class	DiagnosticVerifyCertificateBidirectional			
Note	This meta-class represents the subfunction to do a bidirectional verification of the certificate. **Tags:** atp.recommendedPackage=DiagnosticAuthentications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAuthentication*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.54: DiagnosticVerifyCertificateBidirectional

Class	DiagnosticVerifyCertificateUnidirectional			
Note	This meta-class represents the subfunction to do a unidirectional verification of the certificate. **Tags:** atp.recommendedPackage=DiagnosticAuthentications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAuthentication*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.55: DiagnosticVerifyCertificateUnidirectional

Class	DiagnosticDeAuthentication			
Note	This meta-class represents the subfunction to remove the authentication **Tags:** atp.recommendedPackage=DiagnosticAuthentications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAuthentication*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.56: DiagnosticDeAuthentication

Class	DiagnosticProofOfOwnership			
Note	This meta-class represents the subfunction to provide proof of ownership. **Tags:** atp.recommendedPackage=DiagnosticAuthentications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAuthentication*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.57: DiagnosticProofOfOwnership

Class	DiagnosticAuthTransmitCertificate
Note	This meta-class represents the sub-function to transmit a certificate **Tags:** atp.recommendedPackage=DiagnosticAuthTransmitCertificates
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAuthentication*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*



△

Class	DiagnosticAuthTransmitCertificate			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
certificate Evaluation	DiagnosticAuthTransmit CertificateEvaluation	*	aggr	This aggregation represents a collection of certificate evaluation configurations. **Tags:** atp.Splitkey=certificateEvaluation, certificate Evaluation.shortName

Table 4.58: DiagnosticAuthTransmitCertificate

[TPS_DEXT_01193] Semantics of primitive attribute `DiagnosticAuthTransmitCertificateEvaluation.function` ⌈The attribute `DiagnosticAuthTransmitCertificateEvaluation.function` is used to the define the actual semantics of the corresponding `evaluationId` used for the sub-function transferCertificate (0x04).⌋

[TPS_DEXT_01194] Standardized value of attribute `DiagnosticAuthTransmitCertificateEvaluation.function` ⌈AUTOSAR defines the value **`FUNCTION_SECURE_CODING`** as the only standardized value of attribute `DiagnosticAuthTransmitCertificateEvaluation.function`.⌋

Please note that if it is ensured that a proprietary value of attribute `DiagnosticAuthTransmitCertificateEvaluation.function` does not clash with a potential future extension of the standardized values of attribute `DiagnosticAuthTransmitCertificateEvaluation.function`, it is positively possible to use a value other than the value standardized by [TPS_DEXT_01194].

For example, the avoidance of a potential name clash can be ensured by using a company-specific prefix within the proprietary value.

Class	DiagnosticAuthTransmitCertificateEvaluation			
Note	This meta-class represents the ability to configure a certificate evaluation in the context of a diagnostic authentication.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticAuthTransmitCertificate.certificateEvaluation			
Attribute	Type	Mult.	Kind	Note
evaluationId	PositiveInteger	0..1	attr	This attributes represents the ID of the certificate evaluation.
function	String	0..1	attr	This attribute represents the description of the actual semantics of the corresponding evaluation ID. This Attribute is only used by the AUTOSAR Classic Platform.

Table 4.59: DiagnosticAuthTransmitCertificateEvaluation

4.3.8.4 EcuReset

This chapter describes the modeling of diagnostic services `EcuReset` (0x11).

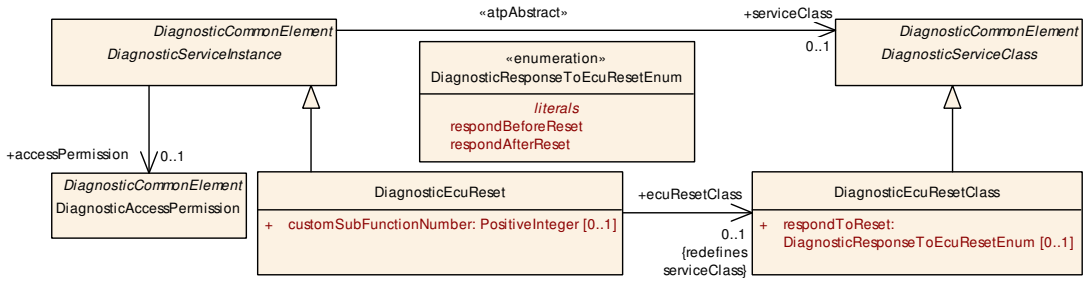

Figure 4.23: Modeling of diagnostic service `EcuReset` (0x11)

Note that (as already explained in Chapter 4.3.8) the `SubFunctions` of this service are modeled by means of the `category` attribute.

[TPS_DEXT_01056] Applicable values for `DiagnosticEcuReset.category` ⌈The following values of the attribute `DiagnosticEcuReset.category` are standardized by AUTOSAR:

- `HARD_RESET`
- `KEY_OFF_ON_RESET`
- `SOFT_RESET`
- `ENABLE_RAPID_POWER_SHUT_DOWN`
- `DISABLE_RAPID_POWER_SHUT_DOWN`

The meaning of these values is described in the applicable [17, ISO 14229-1].⌋

Class	**DiagnosticEcuReset**			
Note	This represents an instance of the "ECU Reset" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticEcuResets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
customSub Function Number	PositiveInteger	0..1	attr	This attribute shall be used to define a custom sub-function number if none of the standardized values of category shall be used.
ecuResetClass	DiagnosticEcuReset Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticEcuReset in the given context. **Tags:** atp.Splitkey=ecuResetClass

Table 4.60: DiagnosticEcuReset

Class	**DiagnosticEcuResetClass**			
Note	This meta-class contains attributes shared by all instances of the "Ecu Reset" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticEcuResets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
respondTo Reset	DiagnosticResponseTo EcuResetEnum	0..1	attr	This attribute defines whether the response to the Ecu Reset service shall be transmitted before or after the actual reset.

Table 4.61: DiagnosticEcuResetClass

Enumeration	**DiagnosticResponseToEcuResetEnum**
Note	This enumeration controls the point in time in which a response to the reception of an EcuReset service shall be generated.
Aggregated by	DiagnosticEcuResetClass.respondToReset
Literal	***Description***
respondAfterReset	Answer to EcuReset service should come after the reset. **Tags:** atp.EnumerationLiteralIndex=0
respondBefore Reset	Answer to EcuReset service should come before the reset. **Tags:** atp.EnumerationLiteralIndex=1

Table 4.62: DiagnosticResponseToEcuResetEnum

[TPS_DEXT_01019] Correspondence of `category` values to numerical values mentioned in the ISO 14229-1 ⌈The [17, ISO 14229-1] standard document defines specific numerical values for the sub-functions of the diagnostic service `EcuReset`.

The correspondence of the numerical values to the pre-defined values of `category` according to [TPS_DEXT_01056] is pretty obvious because the definition of values defined in [TPS_DEXT_01056] has been directly inspired by the [17, ISO 14229-1] standard document.⌋

[TPS_DEXT_01020] Manufacturer-specific values for sub-functions of service `EcuReset` ⌈The [17, ISO 14229-1] standard document, beyond the standardized numerical values for sub-functions, reserves a numerical range of subFunction identifiers for manufacturer- or supplier-specific use.

In this case it is possible to define further values for `category`, provided that a custom prefix is used to avoid potential name clashes with further extensions of the AUTOSAR standard, namely [TPS_DEXT_01056].⌋

[TPS_DEXT_01021] Semantics of `DiagnosticEcuReset.customSubFunctionNumber` ⌈The attribute `DiagnosticEcuReset.customSubFunctionNumber` has been introduced to allow for the specification of a manufacturer- or supplier-specific value to represent the custom sub-function in the diagnostic communication.

The tuple created by the values of attributes `DiagnosticEcuReset.category` and `DiagnosticEcuReset.customSubFunctionNumber` fully specifies identification of the manufacturer- or supplier-specific sub-function.⌋

[constr_1331] Existence of DiagnosticEcuReset.customSubFunctionNumber

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈The attribute DiagnosticEcuReset.customSubFunctionNumber shall only exist if the value of DiagnosticEcuReset.category is outside the standardized set of values as defined by [TPS_DEXT_01056].⌋

[constr_1332] Value range for DiagnosticEcuReset.customSubFunction-Number

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈The allowed value for DiagnosticEcuReset.customSubFunctionNumber shall always be within the closed interval **0x40** .. **0x7E**.⌋

4.3.8.5 CommunicationControl

This chapter describes the modeling of diagnostic services CommunicationControl (0x28). The purpose of this diagnostic service is to enable or disable ISignalIPdu-Groups.

However, the actual implementation of the enabling or disabling is obviously not executed directly within the diagnostic stack. It requires some interaction with the BswM that in turn implements the enabling algorithm.

Therefore, the meta-class modeled for this purpose does not need to refer to ISignalIPduGroups but implements a mode request to the BswM.

[TPS_DEXT_01057] Allowed values of DiagnosticComControl.category ⌈The sub-functions of the diagnostic services CommunicationControl are identified by means of the attribute DiagnosticComControl.category. Standardized values of DiagnosticComControl.category are:

- ENABLE_RX_AND_TX
- DISABLE_RX_AND_TX
- ENABLE_RX_AND_DISABLE_TX
- DISABLE_RX_AND_ENABLE_TX
- ENABLE_RX_AND_DISABLE_TX_WITH_ENHANCED_ADDRESS_INFORMATION
- ENABLE_RX_AND_TX_WITH_ENHANCED_ADDRESS_INFORMATION

The meaning of these values is described in the applicable [17, ISO 14229-1].⌋

[TPS_DEXT_01029] Correspondence of category values to numerical values mentioned in the ISO 14229-1 ⌈The [17, ISO 14229-1] standard document defines specific numerical values for the sub-functions of the diagnostic service CommunicationControl.

The correspondence of the numerical values to the pre-defined values of `category` according to [TPS_DEXT_01057] is pretty obvious because the definition of values defined in [TPS_DEXT_01057] has been directly inspired by the [17, ISO 14229-1] standard document.⌋

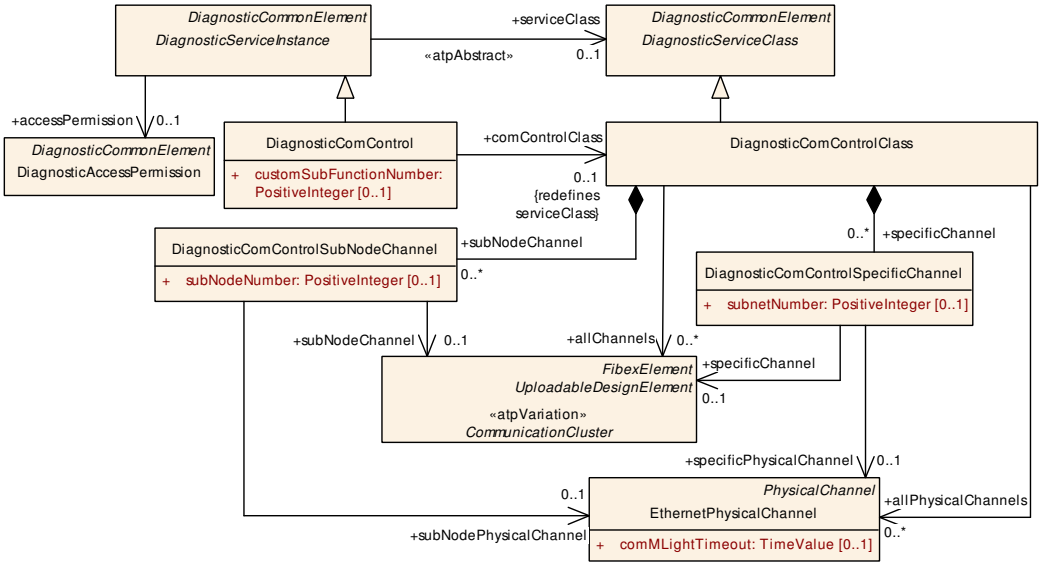

Figure 4.24: Modeling of diagnostic service `CommunicationControl` (0x28)

[TPS_DEXT_01030] Manufacturer-specific values for sub-functions of service `CommunicationControl` ⌈The [17, ISO 14229-1] standard document, beyond the standardized numerical values for sub-functions, reserves a numerical range of sub-Function identifiers for manufacturer-specific use.

In this case it is possible to define further values for `category`, provided that a custom prefix is used to avoid potential name clashes with further extensions of the AUTOSAR standard, namely [TPS_DEXT_01057].⌋

[TPS_DEXT_01031] Semantics of `DiagnosticComControl.customSubFunctionNumber` ⌈The attribute `DiagnosticComControl.customSubFunctionNumber` has been introduced to allow for the specification of a manufacturer-or supplier-specific value to represent the custom sub-function in the diagnostic communication.

The tuple created by the values of attributes `DiagnosticComControl.category` and `DiagnosticComControl.customSubFunctionNumber` fully specifies identification of the manufacturer- or supplier-specific sub-function.⌋

[constr_1334] Existence of `DiagnosticComControl.customSubFunctionNumber`

Imposition time: CP: `IT_DiagDes`, AP: `IT_DiagDes`

⌈The attribute `DiagnosticComControl.customSubFunctionNumber` shall only exist if the value of `DiagnosticComControl.category` is outside the standardized set of values as defined by [TPS_DEXT_01057].⌋

[constr_1335] Possible values for DiagnosticComControl.customSubFunctionNumber

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈Given the fulfillment of [constr_1334], the value of a given DiagnosticComControl.customSubFunctionNumber shall always be within the closed interval **0x40** .. **0x5F** (for manufacturer-specific sub-functions) or the closed interval **0x60** .. **0x7E** (for supplier-specific sub-functions).⌋

[TPS_DEXT_01032] Impact of the DiagnosticComControlClass on the state management for CommunicationClusters ⌈The impact of the DiagnosticComControlClass on the state management for CommunicationClusters can have two alternative consequences:

- All CommunicationClusters are affected. For this purpose the attribute allChannels has the ability to identify the applicable CommunicationClusters.

  It may seem counterintuitive to require a reference to all applicable CommunicationClusters when the expected semantics is actually to define an impact on **all** of them.

  However, there could be private CommunicationClusters that are not participating in the diagnostics work-flow: These need to be kept out of scope and therefore the explicit identification of applicable CommunicationClusters makes sense.

- A selected number of CommunicationClusters is affected. This is conceptually different from the other use case in that it requires an additional attribute that keeps a subnetNumber that is typically assigned by the OEM role.

⌋

In most cases, a reference to the CommunicationCluster would be sufficient as the subject of state management on the respective bus system. In other words, the CommunicationCluster corresponds to a ComM channel.

Currently, the only exception from this rule is Ethernet, where - with respect to communication management - each VLAN (modeled as a EthernetPhysicalChannel) can be managed by an own ComM channel.

As a consequence, the modeling of the DiagnosticComControlClass needs to take the PhysicalChannel into account. On the other hand, the bus systems where a reference to a CommunicationCluster perfectly fits the needs would be better served if the CommunicationCluster is eligible as a reference target.

Therefore, [TPS_DEXT_01032] clarifies the semantics of references to CommunicationCluster while [TPS_DEXT_01186] focuses on the specific role of the EthernetPhysicalChannel in the context of the diagnostic service CommunicationControl.

[TPS_DEXT_01186] Impact of the `DiagnosticComControlClass` on the state management for VLANs ⌈The impact of the `DiagnosticComControlClass` on the state management for VLANs can have two alternative consequences:

- All VLANs are affected. For this purpose, the reference in the role `allPhysicalChannels` has the ability to identify the applicable VLAN.

  It may seem counterintuitive to require a reference to all applicable VLANs when the expected semantics is actually to define an impact on all of them.

  However, there could be private VLANs that are not participating in the diagnostics work-flow: These need to be kept out of scope and therefore the explicit identification of applicable VLANs makes sense.

- A selected number of VLANs is affected. This is conceptually different from the other use case in that it requires an additional attribute that keeps a `subnetNumber` that is typically assigned by the `OEM` role.

⌋

[constr_1336] Applicable value range for `DiagnosticComControlSpecificChannel.subnetNumber`

Imposition time: CP: `IT_DiagDes`, AP: `IT_DiagDes`

⌈The value of attribute `DiagnosticComControlSpecificChannel.subnetNumber` shall be within the closed interval **1** .. **14**.⌋

Class	«atpVariation» ***CommunicationCluster*** (abstract)			
Note	The CommunicationCluster is the main element to describe the topological connection of communicating ECUs. A cluster describes the ensemble of ECUs, which are linked by a communication medium of arbitrary topology (bus, star, ring, ...). The nodes within the cluster share the same communication protocol, which may be event-triggered, time-triggered or a combination of both. A CommunicationCluster aggregates one or more physical channels. **Tags:** vh.latestBindingTime=postBuild			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Subclasses	*AbstractCanCluster*, EthernetCluster, FlexrayCluster, LinCluster, UserDefinedCluster			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
baudrate	PositiveUnlimitedInteger	0..1	attr	Channels speed in bits/s.
comMLight Timeout	TimeValue	0..1	attr	Defines the timeout (in seconds) after COMM_FULL_ COMMUNICATION sub-state COMM_FULL_COM_ READY_SLEEP is left in the ComM channel state machine.

▽

△

Class	«atpVariation» ***CommunicationCluster*** (abstract)			
physical Channel	PhysicalChannel	*	aggr	This relationship defines which channel element belongs to which cluster. A channel shall be assigned to exactly one cluster, whereas a cluster may have one or more channels. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=physicalChannel, physicalChannel.shortName, physicalChannel.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
protocolName	String	0..1	attr	The name of the protocol used.
protocolVersion	String	0..1	attr	The version of the protocol used.

Table 4.63: CommunicationCluster

Please note that the regulation implied by [constr_1336] has not been introduced on an arbitrary basis but gets its conceptual background from the [17, ISO 14229-1] standard document.

Obviously, a diagnostic service with the name `CommunicationControl` will have an impact on the enclosing ECU's mode management. This impact, however, is not defined by any further attributes or references, the `DiagnosticComControl` **is** the impact.

By defining a `DiagnosticComControl` and setting the category to one of the applicable values (e.g. `ENABLE_RX_AND_TX`), it is possible to express the intended semantics to the full extent.

[constr_1337] Allowed value range for attribute `DiagnosticComControlSubNodeChannel.subNodeNumber`

Imposition time: CP: `IT_DiagDes`, AP: `IT_DiagDes`

⌈The value of attribute `DiagnosticComControlSubNodeChannel.subNodeNumber` shall not exceed the closed interval **0** .. **65535**.⌋

[TPS_DEXT_01074] Difference between the attributes `DiagnosticComControlClass.specificChannel` and `DiagnosticComControlClass.subNodeChannel` ⌈The semantical difference between the attributes `DiagnosticComControlClass.specificChannel` and `DiagnosticComControlClass.subNodeChannel` is that `DiagnosticComControlClass.specificChannel` actually refers to a `CommunicationCluster` whereas `DiagnosticComControlClass.subNodeChannel` basically refers to a `CommunicationCluster` to which the nodes with the given identification numbers are connected.⌋

[constr_10122] Existence of the attribute `DiagnosticComControlSubNodeChannel.subNodeChannel`

Imposition time: CP: `IT_DiagDes`, AP: `IT_DiagDes`

⌈Attribute `DiagnosticComControlSubNodeChannel.subNodeChannel` shall only exist if the value of `DiagnosticComControl.category` is set to either

- ENABLE_RX_AND_DISABLE_TX_WITH_ENHANCED_ADDRESS_INFORMATION
  or
- ENABLE_RX_AND_TX_WITH_ENHANCED_ADDRESS_INFORMATION.

⌋

Class	**DiagnosticComControl**			
Note	This represents an instance of the "Communication Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticCommunicationControls			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
comControl Class	DiagnosticComControl Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticComControl in the given context. **Tags:** atp.Splitkey=comControlClass
customSub Function Number	PositiveInteger	0..1	attr	This attribute shall be used to define a custom sub-function number if none of the standardized values of category shall be used.

Table 4.64: DiagnosticComControl

Class	**DiagnosticComControlSpecificChannel**			
Note	This represents the ability to add further attributes to the definition of a specific channel that is subject to the diagnostic service "communication control".			
Base	*ARObject*			
Aggregated by	DiagnosticComControlClass.specificChannel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
specificChannel	CommunicationCluster	0..1	ref	This represents the affected CommunicationCluster in the role specificChannel **Tags:** atp.Splitkey=specificChannel
specificPhysical Channel	EthernetPhysical Channel	0..1	ref	This represents the affected specific EthernetPhysical Channel. **Tags:** atp.Splitkey=specificPhysicalChannel
subnetNumber	PositiveInteger	0..1	attr	This represents the applicable subnet number (which is an arbitrary number ranging from 1..14)

Table 4.65: DiagnosticComControlSpecificChannel

Class	**DiagnosticComControlClass**
Note	This meta-class contains attributes shared by all instances of the "Communication Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticCommunicationControls
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

▽

△

Class	**DiagnosticComControlClass**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
allChannels	CommunicationCluster	*	ref	This reference represents the semantics that all available channels shall be affected. It is still necessary to refer to individual CommunicatuionClusters because there could be private CommunicationClusters in the System Extract that are not subject to the service "communication control". By referring to the applicable CommunicationClusters it can be made sure that only the affected Communication Clusters are accessed. **Tags:** atp.Splitkey=allChannels
allPhysical Channels	EthernetPhysical Channel	*	ref	This reference represents the semantics that all available channels shall be affected. It is still necessary to refer to individual EthernetPhysicalChannels because there could be private VLANs (and thus private EthernetPhysical Channels) in the System Extract that are not subject to the service "communication control". By referring to the applicable EthernetPhysicalChannels it can be made sure that only the affected EthernetPhysical Channels are accessed. **Tags:** atp.Splitkey=allPhysicalChannels xml.namePlural=ALL-PHYSICAL-CHANNELS
specificChannel	DiagnosticComControl SpecificChannel	*	aggr	This represents the ability to add additional attributes to the case that only specific channels are supposed to be considered, **Tags:** atp.Splitkey=specificChannel
subNode Channel	DiagnosticComControl SubNodeChannel	*	aggr	This attribute represents the ability to add further attributes to the definition of a specific sub-node channel that is subject to the diagnostic service "communication control". **Tags:** atp.Splitkey=subNodeChannel

Table 4.66: DiagnosticComControlClass

Class	**DiagnosticComControlSubNodeChannel**			
Note	This represents the ability to add further attributes to the definition of a specific sub-node channel that is subject to the diagnostic service "communication control".			
Base	*ARObject*			
Aggregated by	DiagnosticComControlClass.subNodeChannel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
subNode Channel	CommunicationCluster	0..1	ref	This represents the affected CommunicationCluster in the role subNodeChannel **Tags:** atp.Splitkey=subNodeChannel
subNode Number	PositiveInteger	0..1	attr	This represents the applicable subNode number. The value corresponds to the request message parameter nodeIdentificationNumber of diagnostic service CommunicationControl (0x28).
subNode Physical Channel	EthernetPhysical Channel	0..1	ref	This represents the affected sub-node EthernetPhysical Channel. **Tags:** atp.Splitkey=subNodePhysicalChannel

Table 4.67: DiagnosticComControlSubNodeChannel

4.3.8.6 ControlDTCSetting

This chapter describes the modeling of diagnostic services `ControlDTCSetting` (0x85). The purpose of the diagnostic service is to let the tester tell the diagnostic stack to either stop or resume the updating of a diagnostic trouble code.

[TPS_DEXT_01075] standardized values for the attribute `DiagnosticControlDTCSetting.category` ⌈AUTOSAR does not standardize any of the possible values for the attribute `DiagnosticControlDTCSetting.category`.⌋

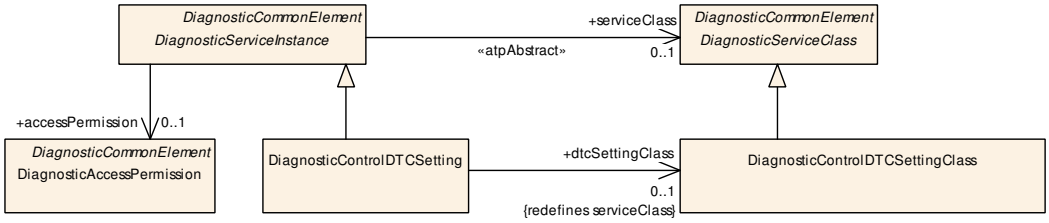

Figure 4.25: Modeling of diagnostic service `ControlDTCSetting` (0x85)

Class	**DiagnosticControlDTCSetting**			
Note	This represents an instance of the "Control DTC Setting" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticControlDtcSettings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dtcSettingClass	DiagnosticControl DTCSettingClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticControlDTCSetting in the given context. **Tags:** atp.Splitkey=dtcSettingClass

Table 4.68: DiagnosticControlDTCSetting

Class	**DiagnosticControlDTCSettingClass**			
Note	This meta-class contains attributes shared by all instances of the "Control DTC Setting" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticControlDtcSettings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.69: DiagnosticControlDTCSettingClass

4.3.8.7 DataByIdentifier

This chapter describes the modeling of diagnostic services ReadDataByIdentifier (0x22) and WriteDataByIdentifier (0x2E).

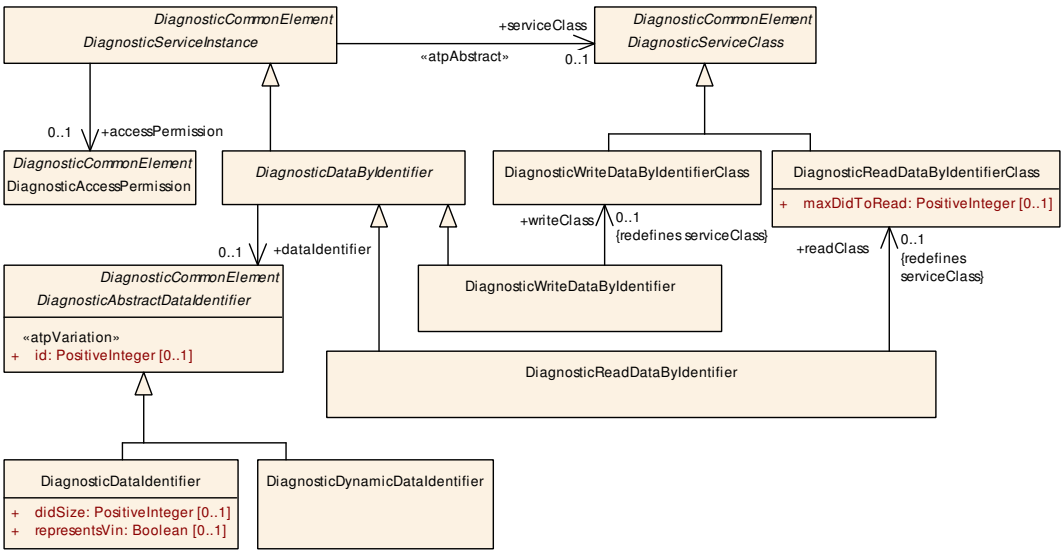

Figure 4.26: Modeling of diagnostic services ReadDataByIdentifier (0x22) and WriteDataByIdentifier (0x2E)

The purpose of this diagnostic service is to enable a tester to request the values of data records from the AUTOSAR diagnostics stack. The data records are identified by a formally modeled DiagnosticDataIdentifier.

The modeling of this diagnostic service comprises the two meta-classes DiagnosticReadDataByIdentifier and DiagnosticWriteDataByIdentifier. These meta-classes both need to specify the set of DiagnosticDataIdentifiers and the set of applicable DiagnosticAccessPermissions.

As these properties are shared between instances of DiagnosticReadDataByIdentifier and DiagnosticWriteDataByIdentifier, an abstract base class named DiagnosticDataByIdentifier has been created that provides the actual references to DiagnosticDataIdentifier and DiagnosticAccessPermission.

The ability to read multiple DIDs at run-time is controlled via attribute DiagnosticReadDataByIdentifierClass.maxDidToRead and therefore it is sufficient to (at configuration-time) limit the multiplicity of attribute dataIdentifier to 1.

Please note that the reference DiagnosticDataByIdentifier.dataIdentifier goes to DiagnosticAbstractDataIdentifier. This modeling approach allows to actually reference any of the meta-classes that inherit from DiagnosticAbstractDataIdentifier.

Class	DiagnosticReadDataByIdentifier			
Note	This represents an instance of the "Read Data by Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDataByIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticDataByIdentifier*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
readClass	DiagnosticReadDataByIdentifierClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticReadDataBy Identifier in the given context. **Tags:** atp.Splitkey=readClass

Table 4.70: DiagnosticReadDataByIdentifier

[constr_1807] Existence of reference DiagnosticDataByIdentifier.dataIdentifier

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticDataByIdentifier, the reference dataIdentifier shall exist.⌋

The modeling of DiagnosticDataByIdentifier represents concrete instances of diagnostic services within a DiagnosticExtract. However, there are attributes that are shared among all instances of DiagnosticReadDataByIdentifier.

For this purpose the dedicated service class DiagnosticReadDataByIdentifierClass has been introduced.

Class	DiagnosticWriteDataByIdentifier			
Note	This represents an instance of the "Write Data by Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDataByIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticDataByIdentifier*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
writeClass	DiagnosticWriteDataByIdentifierClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticWriteDataBy Identifier in the given context. **Tags:** atp.Splitkey=writeClass

Table 4.71: DiagnosticWriteDataByIdentifier

Class	DiagnosticWriteDataByIdentifierClass
Note	This meta-class contains attributes shared by all instances of the "Write Data by Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDataByIdentifiers

▽

△

Class	DiagnosticWriteDataByIdentifierClass			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.72: DiagnosticWriteDataByIdentifierClass

Class	DiagnosticDataByIdentifier (abstract)			
Note	This represents an abstract base class for all diagnostic services that access data by identifier.			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Subclasses	DiagnosticReadDataByIdentifier, DiagnosticReadScalingDataByIdentifier, DiagnosticWriteDataBy Identifier			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataIdentifier	DiagnosticAbstractData Identifier	0..1	ref	This represents the linked DiagnosticDataIdentifier. **Tags:** atp.Splitkey=dataIdentifier

Table 4.73: DiagnosticDataByIdentifier

Class	DiagnosticReadDataByIdentifierClass			
Note	This meta-class contains attributes shared by all instances of the "Read Data by Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDataByIdentifiers			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
maxDidToRead	PositiveInteger	0..1	attr	This attribute represents the maximum number of allowed DIDs in a single instance of DiagnosticReadDataBy Identifier for non-OBD DID requests (at least one DID outside the OBD range of 0xF400-0xF8FF).

Table 4.74: DiagnosticReadDataByIdentifierClass

Please note that it is possible to create a reference to a concrete `DiagnosticData-Identifier` from different `DiagnosticServiceInstance`s.

Class	DiagnosticValueNeeds
Note	Specifies the general needs on the configuration of the Diagnostic Communication Manager (DCM) which are not related to a particular item (e.g. a PID). The main use case is the mapping of service ports to the DCM which are not related to a particular item. In the case of using a sender receiver communicated value, the related value shall be taken via assigned Data in the role "signalBasedDiagnostics". In case of using a client/server communicated value, the related value shall be communicated via the port referenced by assignedPort. The details of this communication (e.g. appropriate naming conventions) are specified in the related software specifications (SWS).

▽

△

Class	**DiagnosticValueNeeds**			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataLength	PositiveInteger	0..1	attr	This attribute is applicable only if the DiagnosticValue Needs is aggregated within a BswModuleDependency. This attribute represents the length of data (in bytes) provided for this particular PID signal. This Attribute is only used by the AUTOSAR Classic Platform.
diagnosticValue Access	DiagnosticValueAccess Enum	0..1	attr	This attribute is applicable only if the DiagnosticValue Needs is aggregated within a BswModuleDependency. This attribute controls whether the data can be read and written or whether it is to be handled read-only.
fixedLength	Boolean	0..1	attr	This attribute is applicable only if the DiagnosticValue Needs is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed.
processingStyle	DiagnosticProcessing StyleEnum	0..1	attr	This attribute controls whether interaction requires the software-component to react synchronously on a request or whether it processes the request in background but still the DCM has to issue the call again to eventually obtain the result of the request.

Table 4.75: DiagnosticValueNeeds

Enumeration	**DiagnosticValueAccessEnum**
Note	Defines the access of the configured diagnostic current values which will be used by the Dem or Dcm module.
Aggregated by	DiagnosticValueNeeds.diagnosticValueAccess
Literal	***Description***
readOnly	The access to the data element is limited to read-only. This is typically used to read-out diagnostic information (e.g. current values). **Tags:** atp.EnumerationLiteralIndex=0
readWrite	The value of the diagnostic data element is classified as configurable (read and write access is possible). **Tags:** atp.EnumerationLiteralIndex=1
writeOnly	The access to the data element is limited to write-only. This supports the use case where the Dcm just writes data to the application software without the intention to read it back, **Tags:** atp.EnumerationLiteralIndex=2

Table 4.76: DiagnosticValueAccessEnum

Enumeration	**DiagnosticProcessingStyleEnum**
Note	This meta-class represents the ability to define the processing style of diagnostic requests.
Aggregated by	DiagnosticValueNeeds.processingStyle
Literal	***Description***

▽

△

Enumeration	**DiagnosticProcessingStyleEnum**
processingStyle Asynchronous	The software-component processes the request in background but still the Dcm has to issue the call again to eventually obtain the result of the request. **Tags:** atp.EnumerationLiteralIndex=0
processingStyle AsynchronousWith Error	The software-component processes the request in background but still the Dcm has to issue the call again to eventually obtain the result of the request or handle error code. **Tags:** atp.EnumerationLiteralIndex=1
processingStyle Synchronous	The software-component is supposed to react synchronously on the request. **Tags:** atp.EnumerationLiteralIndex=2

Table 4.77: DiagnosticProcessingStyleEnum

[TPS_DEXT_01146] Support for service `ReadScalingDataByIdentifier` (0x24) ⌈The AUTOSAR Diagnostic Extract supports the configuration of UDS service `ReadScalingDataByIdentifier` (0x24). For this purpose meta-classes `DiagnosticReadScalingDataByIdentifier` and `DiagnosticReadScalingDataByIdentifierClass` shall be used.⌋

[constr_1623] Restriction on `DiagnosticReadScalingDataByIdentifier.dataIdentifier`

Imposition time: CP: `IT_DiagDes`

⌈The reference `DiagnosticReadScalingDataByIdentifier.dataIdentifier` shall only refer to a `DiagnosticDataIdentifier`.⌋

[constr_1624] Existence of `DiagnosticDataElement.scalingInfoSize`

Imposition time: CP: `IT_DiagDes`

⌈The attribute `DiagnosticDataElement.scalingInfoSize` shall only exist if the enclosing `DiagnosticAbstractParameter` is aggregated by a `DiagnosticDataIdentifier` that is referenced by a `DiagnosticReadScalingDataByIdentifier` in the role `DiagnosticReadScalingDataByIdentifier.dataIdentifier`.⌋

Class	**DiagnosticReadScalingDataByIdentifier**			
Note	This represents an instance of the "Read Scaling Data by Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDataByIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticDataByIdentifier*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	DiagnosticReadScalingDataByIdentifier			
readScaling DataClass	DiagnosticReadScaling DataByIdentifierClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticReadScalingData ByIdentifier in the given context. **Tags:** atp.Splitkey=readScalingDataClass

Table 4.78: DiagnosticReadScalingDataByIdentifier

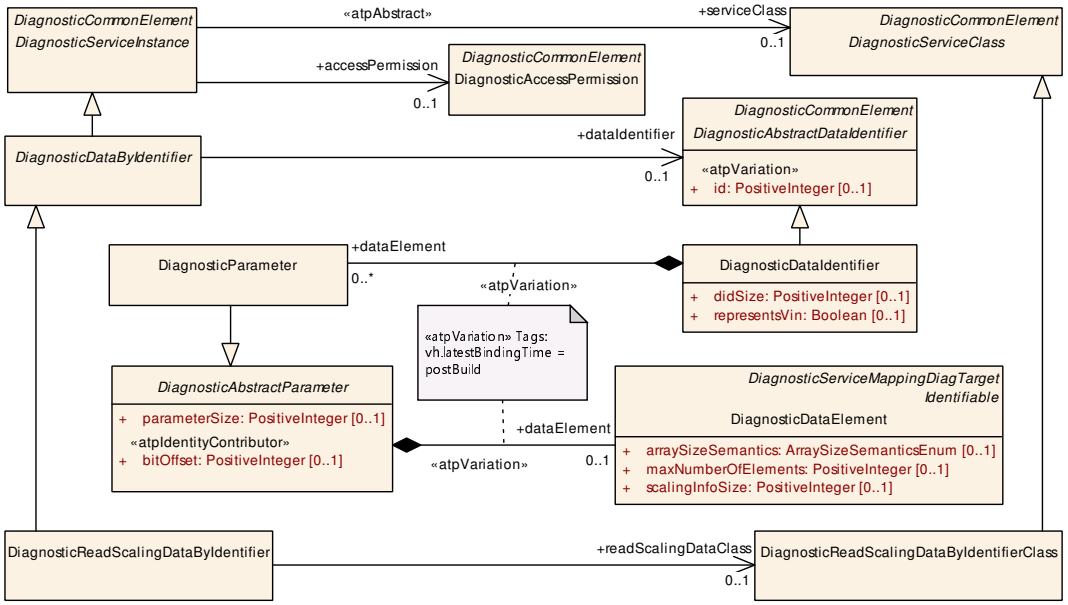

Figure 4.27: Modeling of diagnostic services `ReadScalingDataByIdentifier` (0x24)

Class	DiagnosticReadScalingDataByIdentifierClass			
Note	This meta-class contains attributes shared by all instances of the "Read Scaling Data by Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDataByIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.79: DiagnosticReadScalingDataByIdentifierClass

4.3.8.8 IOControl

This chapter describes the modeling of diagnostic services `InputOutput Control` (0x2F). The purpose of this service is to provide the tester with the ability to override values exchanged with the AUTOSAR hardware abstraction.

[TPS_DEXT_01015] Meaning of attributes of `DiagnosticIOControl` ⌈The attributes `freezeCurrentState`, `resetToDefault`, and `shortTermAdjustment` represent the capabilities of the server rather than a concrete request message.⌋

[TPS_DEXT_01016] The capability returnControlToEcu ⌈According to the statement made by [TPS_DEXT_01015], there is no formal means provided to configure the capability to execute *returnControlToEcu*. This lack of configuration is intentional because the capability is always available and cannot be revoked anyway.⌋

[TPS_DEXT_01017] Meaning of `DiagnosticIOControl.dataIdentifier` ⌈The `DiagnosticIOControl.dataIdentifier` is taken for specifying the payload for the service.

However, in some cases `dataIdentifier` models the payload of the request message (`DiagnosticIOControl.shortTermAdjustment` is set to true) and in some cases it represents the payload of the response message.⌋

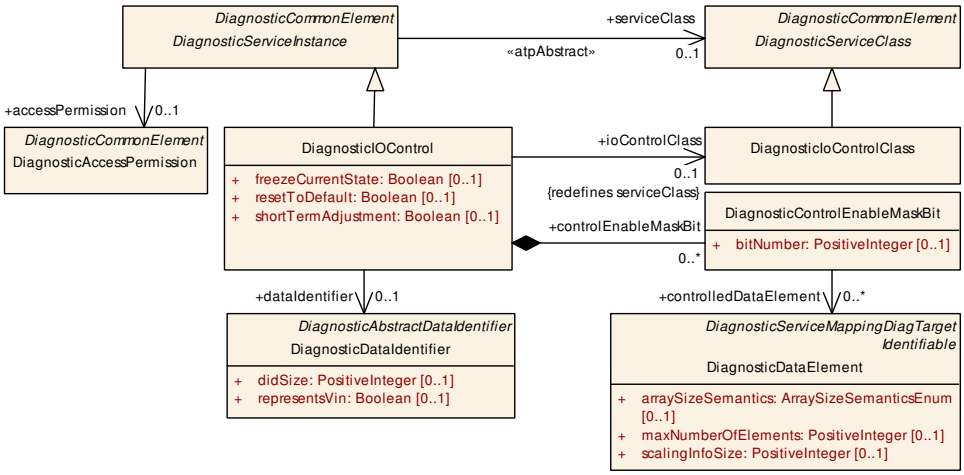

Figure 4.28: Modeling of diagnostic service `IOControl` (0x2F)

Please note that the referenced `dataIdentifier` itself may aggregate several `DiagnosticDataElement`s.

At run-time, only some `DiagnosticDataElement`s may be relevant for a specific execution of the service `InputOutput Control`. For this purpose, the diagnostic message contains the so-called `ControlEnableMaskRecord` (for more information, please refer to [SWS_DCM_00581]).

[TPS_DEXT_01089] Definition of an *identifier* of a `DiagnosticIOControl` ⌈The *identifier* of a `DiagnosticIOControl` is defined by the attribute `DiagnosticIOControl.dataIdentifier.id`.⌋

Class	DiagnosticIOControl			
Note	This represents an instance of the "I/O Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticIoControls			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
controlEnable MaskBit	DiagnosticControl EnableMaskBit	*	aggr	This aggregation represents the control mask record consisting of single bits. **Tags:** atp.Splitkey=controlEnableMaskBit
dataIdentifier	DiagnosticDataIdentifier	0..1	ref	This represents the corresponding DiagnosticData Identifier **Tags:** atp.Splitkey=dataIdentifier
freezeCurrent State	Boolean	0..1	attr	Setting this attribute to true represents the ability of the Dcm to execute a freezeCurrentState.
ioControlClass	DiagnosticIoControl Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticIOControl in the given context. **Tags:** atp.Splitkey=ioControlClass
resetToDefault	Boolean	0..1	attr	Setting this attribute to true represents the ability of the Dcm to execute a resetToDefault.
shortTerm Adjustment	Boolean	0..1	attr	Setting this attribute to true represents the ability of the Dcm to execute a shortTermAdjustment.

Table 4.80: DiagnosticIOControl

Class	DiagnosticIoControlClass			
Note	This meta-class contains attributes shared by all instances of the "IO Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticIoControls			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.81: DiagnosticIoControlClass

[TPS_DEXT_01018] `InputOutput Control` does not define any sub-functions ⌈The diagnostic service `InputOutput Control` does not define any sub-functions, therefore the value of `DiagnosticIOControl.category` does not need to be constrained.⌋

[TPS_DEXT_01150] Semantics of meta-class `DiagnosticControlEnableMaskBit` ⌈By aggregating `DiagnosticControlEnableMaskBit` at `DiagnosticIOControl` it is possible to specify **I/O control channels** in the diagnostic extract.

The reference `DiagnosticControlEnableMaskBit.controlledDataElement` identifies all `DiagnosticDataElement`s that belong to the I/O control channel represented by the bit in the control mask record.⌋

Class	DiagnosticIoControlNeeds			
Note	Specifies the general needs on the configuration of the Diagnostic Communication Manager (DCM) which are not related to a particular item (e.g. a PID). The main use case is the mapping of service ports to the Dcm which are not related to a particular item.			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
currentValue	DiagnosticValueNeeds	0..1	ref	Reference to the DiagnosticValueNeeds indicating the access to the current value via signalBasedDiagnostics. **Tags:** atp.Splitkey=currentValue
freezeCurrent StateSupported	Boolean	0..1	attr	This attribute determines, if the referenced port supports temporary freezing of I/O value.
resetToDefault Supported	Boolean	0..1	attr	This represents a flag for the existence of the ResetTo Default operation in the service interface.
shortTerm Adjustment Supported	Boolean	0..1	attr	This attribute determines, if the referenced port supports temporarily setting of I/O value to a specific value provided by the diagnostic tester.

Table 4.82: DiagnosticIoControlNeeds

Class	DiagnosticControlEnableMaskBit			
Note	This meta-class has the ability to represent one bit in the control enable mask record.			
Base	*ARObject*			
Aggregated by	DiagnosticIOControl.controlEnableMaskBit			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
bitNumber	PositiveInteger	0..1	attr	This attribute represents the bit number of the bit in the control mask record. Bit number 0 is the most significant bit (MSB) in the first byte of the CEMR in the network presentation.
controlledData Element	DiagnosticDataElement	*	ref	This reference represents the collection of Diagnostic DataElements that are controlled by this bit of the control mask record. **Tags:** atp.Splitkey=controlledDataElement

Table 4.83: DiagnosticControlEnableMaskBit

Please note that the modeling of the `controlEnableMaskBit` intentionally does not use the `ordered` annotation to assign bit numbers to the aggregated `DiagnosticControlEnableMaskBit`s.

The control mask record does not necessarily have to be used from top to bottom without gaps.

Example 4.5

Bit 3 could be unused, but 0,1, and 4 could be used.

[constr_1721] `DiagnosticControlEnableMaskBit.bitNumber` shall be unique

Imposition time: CP: `IT_DiagDes`

⌈Within the context of the enclosing `DiagnosticIOControl` the value of attribute `DiagnosticIOControl.controlEnableMaskBit.bitNumber` shall be unique.⌋

Please note that within the context of a DiagnosticIOControl, it is **not** required to have 0 as the lowest value of aggregated DiagnosticIOControl.controlEnableMaskBit.bitNumber.

[constr_1722] Relation between reference DiagnosticIOControl.dataIdentifier and attribute DiagnosticIOControl.controlEnableMaskBit

Imposition time: CP: IT_DiagDes

⌈Any DiagnosticDataElement referenced in the role DiagnosticIOControl.controlEnableMaskBit.controlledDataElement shall be defined in the scope of the DiagnosticDataIdentifier that is referenced in the role DiagnosticIOControl.dataIdentifier.⌋

4.3.8.9 RoutineControl

This chapter describes the modeling of diagnostic services RoutineControl (0x31). The purpose of this diagnostic service is to execute a piece of code, a Diagnostic Routine, on the diagnostic stack at the request of the tester.

Diagnostic Routines consist of up to three possible components:

- Start Routine
- Stop Routine
- Request Routine Results

The impact of this architecture on the meta-model is described by [TPS_DEXT_01077]:

[TPS_DEXT_01077] Modeling of DiagnosticRoutine ⌈From the meta-modeling point of view, the semantics of DiagnosticRoutine is created by aggregating three further meta-classes that represent subfunctions of service RoutineControl (0x31):

- DiagnosticStartRoutine
- DiagnosticStopRoutine
- DiagnosticRequestRoutineResults

⌋

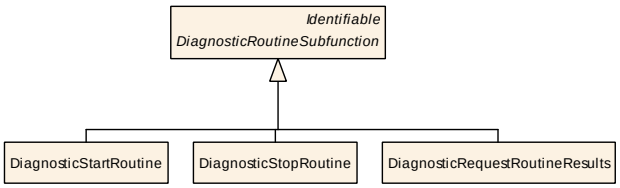

Figure 4.29: Abstract base class for subfunctions of diagnostic service RoutineControl (0x31)

Class	***DiagnosticRoutineSubfunction*** (abstract)			
Note	This meta-class acts as an abstract base class to routine subfunctions.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	DiagnosticRequestRoutineResults, DiagnosticStartRoutine, DiagnosticStopRoutine			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
access Permission	DiagnosticAccess Permission	0..1	ref	This reference represents the access permission of the owning routine subfunction. **Tags:** atp.Splitkey=accessPermission

Table 4.84: DiagnosticRoutineSubfunction

[TPS_DEXT_01088] Semantics of `DiagnosticRoutine.id` ⌈The attribute `DiagnosticRoutine.id` represents the so-called *identifier* of the `DiagnosticRoutine`.⌋

[TPS_DEXT_01078] Not possible to use the attribute `category` for the identification of the sub-function of diagnostic service `RoutineControl` ⌈In the case of `DiagnosticRoutine`, it is not possible to use the attribute `category` for the identification of the sub-function.⌋

The sub-functions actually have different properties i.e. the arguments to a `DiagnosticRoutine` that require a dedicated modeling for this purpose.

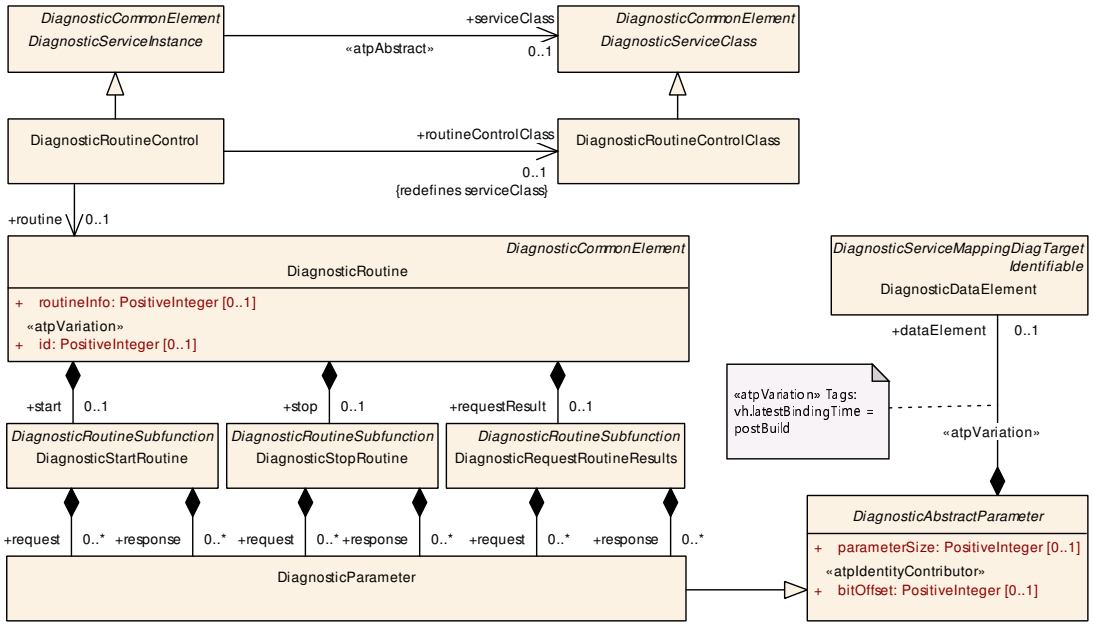

Figure 4.30: Modeling of diagnostic service `RoutineControl` (0x31)

[TPS_DEXT_01079] Modeling of the arguments to a `DiagnosticRoutine` ⌈The arguments to a `DiagnosticRoutine` are modeled by means of `DiagnosticParameter` that is aggregated in the following roles:

- `DiagnosticStartRoutine.request`
- `DiagnosticStartRoutine.response`

- DiagnosticStopRoutine.request
- DiagnosticStopRoutine.response
- DiagnosticRequestRoutineResults.request
- DiagnosticRequestRoutineResults.response

⌋

A DiagnosticParameter, in turn, aggregates a DiagnosticDataElement (see Figure 4.1) in the role dataElement.

[TPS_DEXT_01080] Diagnostic Routine needs to be started ⌈The standard document [17, ISO 14229-1] does not foresee the existence of a Diagnostic Routine that is already executing at boot time. Therefore, a Diagnostic Routine needs to be started at some point in order to make sense of it.⌋

[constr_1339] Existence of DiagnosticRoutine.start

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈In a complete DiagnosticExtract, the attribute DiagnosticRoutine.start shall always exist for any given DiagnosticRoutine.⌋

[TPS_DEXT_01035] Existence of DiagnosticRoutine.stop and DiagnosticRoutine.requestResult ⌈In contrast to DiagnosticRoutine.start (as clarified by [constr_1339]), the existence of DiagnosticRoutine.stop and DiagnosticRoutine.requestResult is truly optional.⌋

[constr_1340] Consistency of DiagnosticServiceSwMapping with respect to synchronously called DiagnosticRoutines

Imposition time: CP: IT_DiagDes

⌈Each DiagnosticServiceSwMapping that references a DiagnosticRoutineControl that only aggregates a DiagnosticStartRoutine in the role start shall only reference a SwcServiceDependency or BswServiceDependency that in turn aggregates a DiagnosticRoutineNeeds with attribute diagRoutineType set to DiagnosticRoutineTypeEnum.synchronous.⌋

[constr_1341] Consistency of DiagnosticServiceSwMapping with respect to asynchronously called DiagnosticRoutines

Imposition time: CP: IT_DiagDes

⌈Each DiagnosticServiceSwMapping that references a DiagnosticRoutineControl that aggregates a DiagnosticStopRoutine and/or DiagnosticRequestRoutineResults in the role stop or requestResult shall only reference a SwcServiceDependency or BswServiceDependency that in turn aggregates a DiagnosticRoutineNeeds with attribute diagRoutineType set to DiagnosticRoutineTypeEnum.asynchronous.⌋

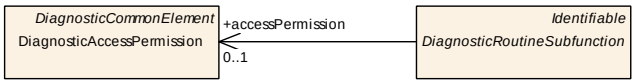

Figure 4.31: Permission is granted per subfunction

The DiagnosticRoutineControl service is the only UDS service with both sub-function and identifier. The [17, ISO 14229-1] treats the service with respect to session and security as a service with identifier.

In other words a DiagnosticRoutine—with its three subfunctions start, stop and requestResult—for the same identifier can have only DiagnosticAccess-Permission with the same referenced DiagnosticSession and Diagnostic-SecurityLevel.

However, different DiagnosticEnvironmentalConditions may exist for the subfunctions of a DiagnosticRoutine with the same identifier (Diagnostic-Routine.id).

[constr_1772] Unique DiagnosticSession and DiagnosticSecurityLevel for diagnostic routines that have the same identifier

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈All DiagnosticAccessPermissions referenced from DiagnosticRoutines where attribute DiagnosticRoutine.id has the identical value shall refer to the identical set of DiagnosticSession and DiagnosticSecurityLevel.⌋

For this purpose, the reference from the abstract base class[7] DiagnosticRoutine-Subfunction to DiagnosticAccessPermission in the role accessPermission exists (see Figure 4.31).

Consequently, the reference from meta-class DiagnosticRoutineControl (via its abstract base class DiagnosticServiceInstance) to meta-class Diagnostic-AccessPermission has no meaning.

[TPS_DEXT_01196] Reference from DiagnosticRoutineControl to DiagnosticAccessPermission has no meaning ⌈If the reference Diagnostic-RoutineControl.accessPermission (via its abstract base class Diagnostic-ServiceInstance) exists, then the value of the reference in the role access-Permission shall be ignored.⌋

Rationale: for the execution of routines, the access permission needs to be defined on the level of sub-functions.

[7] of meta-classes DiagnosticStartRoutine, DiagnosticStopRoutine, and Diagnostic-RequestRoutineResults

Class	DiagnosticRoutine			
Note	This meta-class represents the ability to define a diagnostic routine. **Tags:** atp.recommendedPackage=DiagnosticRoutines			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
id	PositiveInteger	0..1	attr	This is the numerical identifier used to identify the DiagnosticRoutine in the scope of diagnostic workflow **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
requestResult	DiagnosticRequestRoutineResults	0..1	aggr	This represents the ability to request the result of a running routine. **Tags:** atp.Splitkey=requestResult, requestResult.shortName
routineInfo	PositiveInteger	0..1	attr	This represents the routine info byte. The info byte contains a manufacturer-specific value (for the identification of record identifiers) that is reported to the tester. Other use cases for this attribute are mentioned in ISO 27145 and ISO 26021.
start	DiagnosticStartRoutine	0..1	aggr	This represents the ability to start a routine **Tags:** atp.Splitkey=start, start.shortName
stop	DiagnosticStopRoutine	0..1	aggr	This represents the ability to stop a running routine. **Tags:** atp.Splitkey=stop, stop.shortName

Table 4.85: DiagnosticRoutine

[constr_1815] Existence of attribute `DiagnosticRoutine.id`

Imposition time: CP: `IT_DiagDes`, AP: `IT_DiagDes`

⌈For each `DiagnosticRoutine`, the attribute `id` shall exist at least once.⌋

Class	DiagnosticStartRoutine			
Note	This represents the ability to start a diagnostic routine.			
Base	*ARObject*, *DiagnosticRoutineSubfunction*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticRoutine.start			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
request	DiagnosticParameter	*	aggr	This represents the request parameters. **Tags:** atp.Splitkey=request, request.bitOffset, request.ident.shortName
response	DiagnosticParameter	*	aggr	This represents the response parameters. **Tags:** atp.Splitkey=response, response.bitOffset, response.ident.shortName

Table 4.86: DiagnosticStartRoutine

Class	DiagnosticStopRoutine			
Note	This represents the ability to stop a diagnostic routine.			
Base	*ARObject*, *DiagnosticRoutineSubfunction*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticRoutine.stop			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
request	DiagnosticParameter	*	aggr	This represents the request parameters. **Tags:** atp.Splitkey=request, request.bitOffset, request.ident.shortName
response	DiagnosticParameter	*	aggr	This represents the response parameters. **Tags:** atp.Splitkey=response, response.bitOffset, response.ident.shortName

Table 4.87: DiagnosticStopRoutine

Class	DiagnosticRequestRoutineResults			
Note	This meta-class represents the ability to define the result of a diagnostic routine execution.			
Base	*ARObject*, *DiagnosticRoutineSubfunction*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticRoutine.requestResult			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
request	DiagnosticParameter	*	aggr	This represents the request parameters. **Tags:** atp.Splitkey=request, request.bitOffset, request.ident.shortName
response	DiagnosticParameter	*	aggr	This represents the response parameters. **Tags:** atp.Splitkey=response, response.bitOffset, response.ident.shortName

Table 4.88: DiagnosticRequestRoutineResults

Class	DiagnosticRoutineControl			
Note	This represents an instance of the "Routine Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRoutineControls			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
routine	DiagnosticRoutine	0..1	ref	This refers to the applicable DiagnosticRoutine. **Tags:** atp.Splitkey=routine
routineControlClass	DiagnosticRoutineControlClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRoutineControl in the given context. **Tags:** atp.Splitkey=routineControlClass

Table 4.89: DiagnosticRoutineControl

[constr_10100] Existence of `DiagnosticRoutineControl.routine`

Imposition time: CP: `IT_DiagDes`, AP: `IT_DiagDes`

⌈For each `DiagnosticRoutineControl`, the attribute `routine` shall exist.⌋

Class	DiagnosticRoutineControlClass			
Note	This meta-class contains attributes shared by all instances of the "Routine Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRoutineControls			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.90: DiagnosticRoutineControlClass

Class	DiagnosticRoutineNeeds			
Note	Specifies the general needs on the configuration of the Diagnostic Communication Manager (Dcm) which are not related to a particular item (e.g. a PID). The main use case is the mapping of service ports to the Dcm which are not related to a particular item.			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagRoutine Type	DiagnosticRoutineType Enum	0..1	attr	This denotes the type of diagnostic routine which is implemented by the referenced server port.

Table 4.91: DiagnosticRoutineNeeds

Enumeration	DiagnosticRoutineTypeEnum
Note	This enumerator specifies the different types of diagnostic routines.
Aggregated by	DiagnosticRoutineNeeds.diagRoutineType
Literal	**Description**
asynchronous	This indicates that the diagnostic server is not blocked while the diagnostic routine is running. **Tags:** atp.EnumerationLiteralIndex=0
synchronous	This indicates that the diagnostic routine blocks the diagnostic server in the ECU while the routine is running. **Tags:** atp.EnumerationLiteralIndex=1

Table 4.92: DiagnosticRoutineTypeEnum

4.3.8.10 DynamicallyDefineDataIdentifier

This chapter describes the modeling of diagnostic services `DynamicallyDefineDataIdentifier` (0x2C). The purpose of the service is to allow for defining data identifiers (`DID`) at run-time.

By this means it is possible to combine existing diagnostic data into a single DID.

This semantics is reflected by the modeling of the meta-class `DiagnosticDynamicallyDefineDataIdentifier` that refers to a `DiagnosticDynamicDataIdentifier` in the role `dataIdentifier`.

Also, the `DiagnosticDynamicallyDefineDataIdentifier` inherits a reference to `accessPermission` from `DiagnosticServiceInstance`.

[constr_1421] Consistency of attribute DiagnosticDynamicallyDefineData-IdentifierClass.subfunction

Imposition time: CP: IT_DiagDes

⌈The values of DiagnosticDynamicallyDefineDataIdentifierClass.subfunction shall not repeat, i.e. every value of DiagnosticDynamicallyDefine-DataIdentifierSubfunctionEnum shall at most appear once in the subfunction attribute.⌋

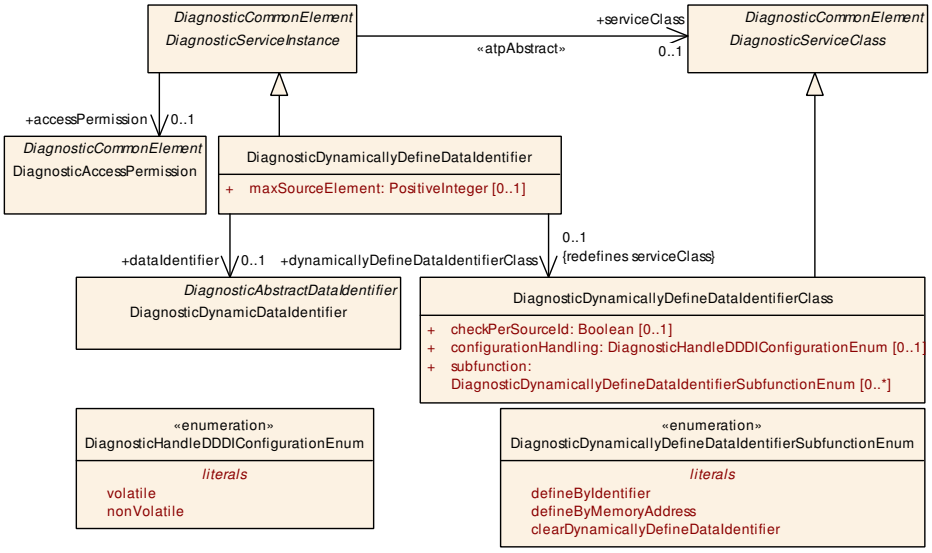

Figure 4.32: Modeling of diagnostic service DynamicallyDefineDataIdentifier (0x2C)

Class	**DiagnosticDynamicallyDefineDataIdentifier**			
Note	This represents an instance of the "Dynamically Define Data Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDynamicallyDefineDataIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataIdentifier	DiagnosticDynamicData Identifier	0..1	ref	This represents the applicable DiagnosticDynamicData Identfier. **Tags:** atp.Splitkey=dataIdentifier
dynamically DefineData IdentifierClass	DiagnosticDynamically DefineDataIdentifier Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticDynamicallyDefine DataIdentifier in the given context. **Tags:** atp.Splitkey=dynamicallyDefineDataIdentifierClass
maxSource Element	PositiveInteger	0..1	attr	This represents the maximum number of source elements of the dynamically created DID.

Table 4.93: DiagnosticDynamicallyDefineDataIdentifier

[constr_1808] Existence of reference DiagnosticDynamicallyDefineData-Identifier.dataIdentifier

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticDynamicallyDefineDataIdentifier, the reference to DiagnosticDynamicDataIdentifier in the role dataIdentifier shall exist.⌋

Class	**DiagnosticDynamicallyDefineDataIdentifierClass**			
Note	This meta-class contains attributes shared by all instances of the "Dynamically Define Data Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDynamicallyDefineDataIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
checkPer SourceId	Boolean	0..1	attr	If set to TRUE, the Dcm module shall check the session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22) with DID in the range 0x F200 to 0xF3FF. If set to FALSE. the Dcm module shall not check the session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF.
configuration Handling	DiagnosticHandle DDDIConfiguration Enum	0..1	attr	This configuration switch defines whether DDDID definition is handled as non-volatile information or not.
subfunction	DiagnosticDynamically DefineDataIdentifier SubfunctionEnum	*	attr	This attribute contains a list of applicable subfunctions for all DiagnosticDynamicallyDefineDataIdentifier that reference the DiagnosticDynamicallyDefineDataIdentifier Class.

Table 4.94: DiagnosticDynamicallyDefineDataIdentifierClass

Enumeration	**DiagnosticHandleDDDIConfigurationEnum**
Note	This meta-class represents the options for controlling how the configuration of the DynamicallyDefine DataIdentifiers is done in the given context.
Aggregated by	DiagnosticDynamicallyDefineDataIdentifierClass.configurationHandling
Literal	***Description***
nonVolatile	This indicates that the configuration of DynamicallyDefineDataIdentifier shall be stored as non-volatile data. **Tags:** atp.EnumerationLiteralIndex=0
volatile	This indicates that the configuration of DynamicallyDefineDataIdentifier shall be handled as volatile data. **Tags:** atp.EnumerationLiteralIndex=1

Table 4.95: DiagnosticHandleDDDIConfigurationEnum

Enumeration	**DiagnosticDynamicallyDefineDataIdentifierSubfunctionEnum**
Note	This meta-class contains a list of possible subfunctions for the UDS service 0x2C.
Aggregated by	DiagnosticDynamicallyDefineDataIdentifierClass.subfunction
Literal	***Description***

▽

△

Enumeration	**DiagnosticDynamicallyDefineDataIdentifierSubfunctionEnum**
clearDynamically DefineDataIdentifier	Clear the specified dynamic data identifier. **Tags:** atp.EnumerationLiteralIndex=0
defineByIdentifier	The definition of dynamic data identifier shall be done via a reference to a diagnostic data identifier. **Tags:** atp.EnumerationLiteralIndex=1
defineByMemory Address	The definition of dynamic data identifier shall be done via a reference to a memory address. **Tags:** atp.EnumerationLiteralIndex=2

Table 4.96: DiagnosticDynamicallyDefineDataIdentifierSubfunctionEnum

4.3.8.11 ReadDataByPeriodicIdentifier

This chapter describes the modeling of diagnostic services `ReadDataByPeriodicIdentifier` (0x2A).

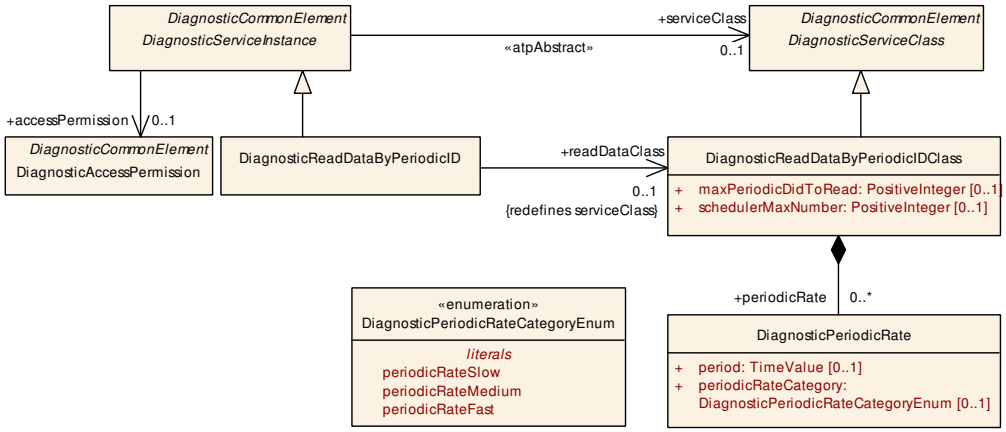

Figure 4.33: Modeling of diagnostic service `ReadDataByPeriodicIdentifier` (0x2A)

[constr_1338] Maximum number of aggregated `DiagnosticReadDataByPeriodicIDClass.periodicRate`

Imposition time: CP: `IT_DiagDes`

⌈The number of aggregated `periodicRate` within the context of one `DiagnosticReadDataByPeriodicIDClass` shall be within the closed interval 1..3.⌋

Class	**DiagnosticReadDataByPeriodicID**			
Note	This represents an instance of the "Read Data by periodic Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticReadDataByPeriodicIds			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**DiagnosticReadDataByPeriodicID**			
readDataClass	DiagnosticReadDataBy PeriodicIDClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticReadDataBy PeriodicID in the given context. **Tags:** atp.Splitkey=readDataClass

Table 4.97: DiagnosticReadDataByPeriodicID

Class	**DiagnosticReadDataByPeriodicIDClass**			
Note	This meta-class contains attributes shared by all instances of the "Read Data by periodic Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticReadDataByPeriodicIds			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
maxPeriodicDid ToRead	PositiveInteger	0..1	attr	This represents the maximum number of data identifiers that can be included in one request.
periodicRate	DiagnosticPeriodicRate	*	aggr	This represents the description of a collection of periodic rates in which the service can be executed. **Tags:** atp.Splitkey=periodicRate
schedulerMax Number	PositiveInteger	0..1	attr	This represents the maximum number of periodic data identifiers that can be scheduled in parallel.

Table 4.98: DiagnosticReadDataByPeriodicIDClass

[constr_1810] Existence of the aggregation DiagnosticReadDataByPeriodic-IDClass.periodicRate

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticReadDataByPeriodicIDClass, the aggregation of DiagnosticPeriodicRate in the role periodicRate shall exist at least once.⌋

[constr_1811] Existence of the attribute DiagnosticReadDataByPeriodic-IDClass.maxPeriodicDidToRead

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticReadDataByPeriodicIDClass, the attribute max-PeriodicDidToRead shall exist at least once.⌋

[constr_1812] Existence of the attribute DiagnosticReadDataByPeriodic-IDClass.schedulerMaxNumber

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticReadDataByPeriodicIDClass, the attribute scheduler-MaxNumber shall exist at least once.⌋

Class	**DiagnosticPeriodicRate**			
Note	This represents the ability to define a periodic rate for the specification of the "read data by periodic ID" diagnostic service.			
Base	*ARObject*			
Aggregated by	DiagnosticReadDataByPeriodicIDClass.periodicRate			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
period	TimeValue	0..1	attr	This represents the period of the DiagnosticPeriodicRate in seconds.
periodicRate Category	DiagnosticPeriodicRate CategoryEnum	0..1	attr	This attribute represents the category of the periodic rate.

Table 4.99: DiagnosticPeriodicRate

[constr_1763] Existence of attribute DiagnosticPeriodicRate.periodicRateCategory

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticPeriodicRate, the attribute periodicRateCategory shall exist.⌋

Enumeration	**DiagnosticPeriodicRateCategoryEnum**
Note	This meta-class provides possible values for the setting of the periodic rate.
Aggregated by	DiagnosticPeriodicRate.periodicRateCategory
Literal	***Description***
periodicRateFast	This value represents a fast periodic rate. **Tags:** atp.EnumerationLiteralIndex=0
periodicRate Medium	This value represents a medium periodic rate. **Tags:** atp.EnumerationLiteralIndex=1
periodicRateSlow	This value represents a slow periodic rate. **Tags:** atp.EnumerationLiteralIndex=2

Table 4.100: DiagnosticPeriodicRateCategoryEnum

The meaning of the values defined in DiagnosticPeriodicRateCategoryEnum is described in the applicable [17, ISO 14229-1].

[TPS_DEXT_01202] Applicability of DiagnosticReadDataByPeriodicID

⌈The service 0x2A, formalized by meta-class DiagnosticReadDataByPeriodicID can transmit specific DiagnosticDataIdentifier in the enclosing DiagnosticContributionSet.

In particular, the DiagnosticDataIdentifiers that can be used in combination with a DiagnosticReadDataByPeriodicID need to fulfill the following conditions:

attribute id is within the range for periodicDataIdentifiers (technical term defined in the UDS standard).

The DiagnosticDataIdentifier is referenced by a DiagnosticReadDataByIdentifier in the role DiagnosticDataByIdentifier.dataIdentifier.

As a consequence, the DiagnosticAccessPermission associated with the individual DiagnosticDataIdentifiers (that fulfill the condition described above) applies.⌋

[constr_10752] Existence of DiagnosticReadDataByPeriodicID ⌈In the context of any DiagnosticContributionSet, at most one DiagnosticReadDataByPeriodicID can be defined.⌋

4.3.8.12 ResponseOnEvent

This chapter describes the modeling of diagnostic services ResponseOnEvent (0x86). The purpose of this service is to instruct the AUTOSAR diagnostic stack with respect to the starting or stopping of sending responses to a specific event to the tester.

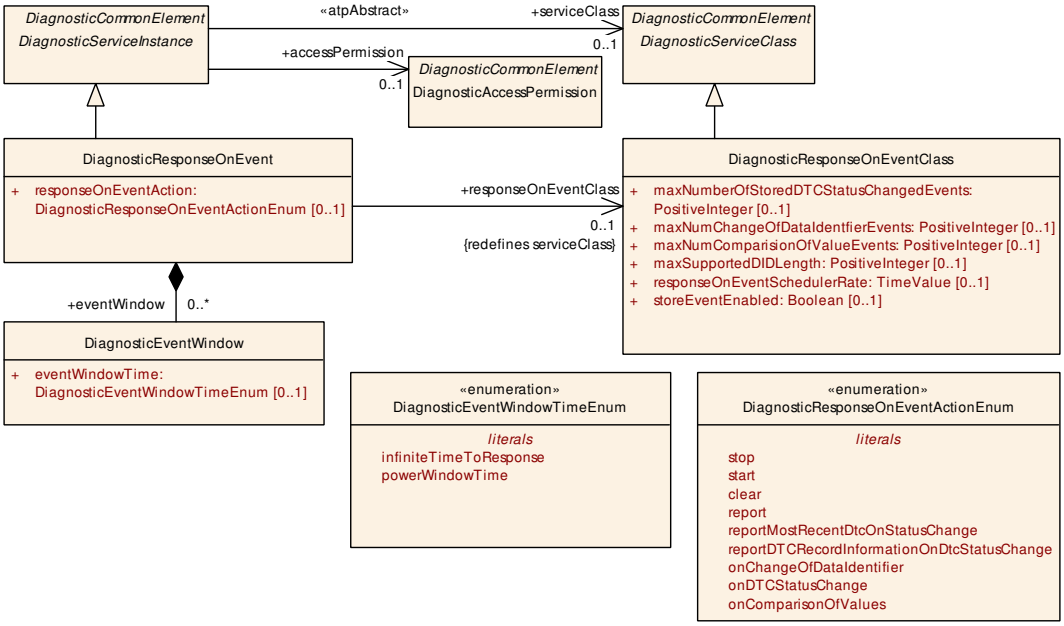

Figure 4.34: Modeling of diagnostic service ResponseOnEvent (0x86)

Class	**DiagnosticResponseOnEvent**			
Note	This represents an instance of the "Response on Event" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticResponseOnEvents			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
eventWindow	DiagnosticEventWindow	*	aggr	This represents the applicable DiagnosticEventWindows **Tags:** atp.Splitkey=eventWindow
responseOn EventAction	DiagnosticResponseOn EventActionEnum	0..1	attr	Defines sub-functions of the service ResponseOnEvent.

▽

△

Class	DiagnosticResponseOnEvent			
responseOn EventClass	DiagnosticResponseOn EventClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticResponseOnEvent in the given context. **Tags:** atp.Splitkey=responseOnEventClass

Table 4.101: DiagnosticResponseOnEvent

Class	DiagnosticResponseOnEventClass			
Note	This represents the ability to define common properties for all instances of the "Response on Event" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticResponseOnEvents			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
maxNumberOf Stored DTCStatus ChangedEvents	PositiveInteger	0..1	attr	The maximum number of DTCs that can be stored as DTCs with change status within one ResponseOnEvent SchedulerRate interval.
maxNum ChangeOfData IdentfierEvents	PositiveInteger	0..1	attr	The maximum number of events that can be simultaneously configured with sub function onChangeOf DataIdentifier.
maxNum ComparisionOf ValueEvents	PositiveInteger	0..1	attr	The maximum number of events that can be simultaneously configured with sub function on ComparisonOfValues.
maxSupported DIDLength	PositiveInteger	0..1	attr	The maximum number of measurable data bytes allowed for each DID that is used for comparison or data change.
responseOn EventScheduler Rate	TimeValue	0..1	attr	The call rate of the periodic scheduler to compare the values of the DataIdentifier (DID) or to detect DTC status changes.
storeEvent Enabled	Boolean	0..1	attr	Specifies if the storeEvent functionality of the Response OnEvent diagnostic service shall be supported or not. If set to true, the storeEvent functionality is available. If set to false the storeEvent functionality is not available.

Table 4.102: DiagnosticResponseOnEventClass

[constr_10577] Existence of DiagnosticResponseOnEventClass.storeEventEnabled

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticResponseOnEventClass, the attribute storeEventEnabled shall exist.⌋

Class	DiagnosticEventWindow
Note	This represents the ability to define the characteristics of the applicable event window
Base	*ARObject*
Aggregated by	DiagnosticResponseOnEvent.eventWindow

▽

△

Class	**DiagnosticEventWindow**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
eventWindow Time	DiagnosticEventWindow TimeEnum	0..1	attr	This attribute clarifies the validity of the eventWindow

Table 4.103: DiagnosticEventWindow

Enumeration	**DiagnosticEventWindowTimeEnum**
Note	This represents the ability to define the semantics of the event window.
Aggregated by	DiagnosticEventWindow.eventWindowTime
Literal	***Description***
infiniteTimeTo Response	This value specifies that the event window shall stay active for an infinite amount of time (e.g. open window until power off). **Tags:** atp.EnumerationLiteralIndex=3
powerWindowTime	This enumeration value specifies that the server shall send response on event messages until the server is powered down. The server stops sending response on event messages with the power down and will send no more response on event messages after server is up again. **Tags:** atp.EnumerationLiteralIndex=4

Table 4.104: DiagnosticEventWindowTimeEnum

Enumeration	**DiagnosticResponseOnEventActionEnum**
Note	This meta-class has the ability to define sub-functions of the UDS service ResponseOnEvent.
Aggregated by	DiagnosticResponseOnEvent.responseOnEventAction
Literal	***Description***
clear	Clears the configured events. **Tags:** atp.EnumerationLiteralIndex=2
onChangeOfData Identifier	Reports based on change of data identifier. **Tags:** atp.EnumerationLiteralIndex=6
onComparisonOf Values	Triggered if data condition is met (e.g. RPM over 5000 1/min). **Tags:** atp.EnumerationLiteralIndex=8
onDTCStatus Change	Reports based on change of DTC status. **Tags:** atp.EnumerationLiteralIndex=7
report	Reports the activated events. **Tags:** atp.EnumerationLiteralIndex=3
reportDTCRecord InformationOnDtc StatusChange	Reports the DTC record-related information based on a DTC status change. (Subfunction 0x09) **Tags:** atp.EnumerationLiteralIndex=5
reportMostRecent DtcOnStatus Change	Triggers the report of the most recent failed or confirmed DTC (Subfunction 0x08). **Tags:** atp.EnumerationLiteralIndex=4
start	Starts the response on event service. **Tags:** atp.EnumerationLiteralIndex=1
stop	Stops the response on event service. **Tags:** atp.EnumerationLiteralIndex=0

Table 4.105: DiagnosticResponseOnEventActionEnum

4.3.8.13 ReadDTCInformation

This chapter describes the modeling of diagnostic services `ReadDTCInformation` (0x19). The purpose of this service is to enable a tester to read a `Diagnostic Trouble Code` from the AUTOSAR `Dcm`, see [10, SWS Diagnostic Communication Manager] (that, in turn, fetches the information from the AUTOSAR `Dem`, see [11, SWS Diagnostic Event Manager]).

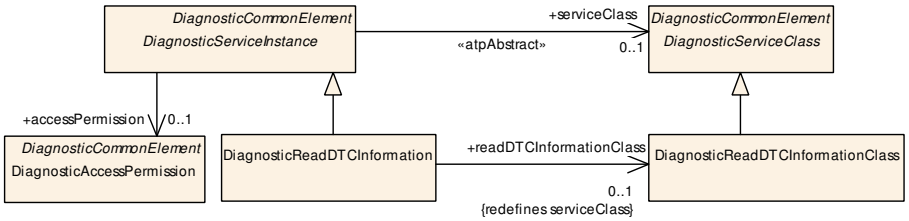

Figure 4.35: Modeling of diagnostic service `ReadDTCInformation` (0x19)

[TPS_DEXT_01034] Sub-functions of the service `ReadDTCInformation` ⌈The diagnostics service `ReadDTCInformation` defines a number of sub-functions that are, as far as the modeling in AUTOSAR goes, identified by a textual identifier.

These sub-functions are **not** modeled explicitly but can be specified by defining a `DiagnosticReadDTCInformation` and by setting the `DiagnosticReadDTCInformation.category` to the identifier of the respective sub-function.

The possible values, as far as the AUTOSAR standard is concerned, are defined by [TPS_DEXT_01060].⌋

[TPS_DEXT_01060] Applicable values for primitive attribute `DiagnosticReadDTCInformation.category` ⌈The following values of the attribute `DiagnosticReadDTCInformation.category` are standardized by AUTOSAR:

- `REPORT_NUMBER_OF_DTC_BY_STATUS_MASK`
- `REPORT_DTC_BY_STATUS_MASK`
- `REPORT_NUMBER_OF_EMISSIONS_OBD_DTC_BY_STATUS_MASK`
- `REPORT_EMISSIONS_OBD_DTC_BY_STATUS_MASK`
- `REPORT_DTC_SNAPSHOT_IDENTIFICATION`
- `REPORT_DTC_SNAPSHOT_RECORD_BY_DTC_NUMBER`
- `REPORT_DTC_STORED_DATA_BY_RECORD_NUMBER`
- `REPORT_DTC_EXT_DATA_RECORD_BY_DTC_NUMBER`
- `REPORT_NUMBER_OF_DTC_BY_SEVERITY_MASK_RECORD`
- `REPORT_DTC_BY_SEVERITY_MASK_RECORD`
- `REPORT_SEVERITY_INFORMATION_OF_DTC`

- REPORT_SUPPORTED_DTC
- REPORT_FIRST_TEST_FAILED_DTC
- REPORT_FIRST_CONFIRMED_DTC
- REPORT_MOST_RECENT_TEST_FAILED_DTC
- REPORT_MOST_RECENT_CONFIRMED_DTC
- REPORT_DTC_FAULT_DETECTION_COUNTER
- REPORT_DTC_WITH_PERMANENT_STATUS
- REPORT_DTC_EXT_DATA_RECORD_BY_RECORD_NUMBER
- REPORT_USER_DEF_MEMORY_DTC_BY_STATUS_MASK
- REPORT_USER_DEF_MEMORY_DTC_SNAPSHOT_RECORD_BY_DTC_NUMBER
- REPORT_USER_DEF_MEMORY_DTC_EXT_DATA_RECORD_BY_DTC_NUMBER
- REPORT_WWH_OBD_DTC_BY_MASK_RECORD
- REPORT_WWH_OBD_DTC_WITH_PERMANENT_STATUS
- REPORT_SUPPORTED_DTC_EXT_DATA_RECORD
- REPORT_DTC_INFORMATION_BY_DTC_READINESS_GROUP_IDENTIFIER

The meanings of these values are described in the applicable [17, ISO 14229-1].⌋

Please note that there is nothing to configure for DiagnosticReadDTCInformation beyond its mere existence.

Class	**DiagnosticReadDTCInformation**			
Note	This represents an instance of the "Read DTC Information" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticReadDtcInformations			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
read DTCInformation Class	DiagnosticRead DTCInformationClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRead DTCInformation in the given context. **Tags:** atp.Splitkey=readDTCInformationClass

Table 4.106: DiagnosticReadDTCInformation

Class	**DiagnosticReadDTCInformationClass**			
Note	This meta-class contains attributes shared by all instances of the "ReadDTCInformation" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticReadDtcInformations			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.107: DiagnosticReadDTCInformationClass

4.3.8.14 ClearDiagnosticInformation

This chapter describes the modeling of diagnostic services `ClearDiagnosticInformation` (0x14). As the name suggests, the purpose of the service is to clear diagnostic information in the AUTOSAR diagnostics stack.

Please note that there is nothing to configure for `DiagnosticClearDiagnosticInformation` beyond its mere existence.

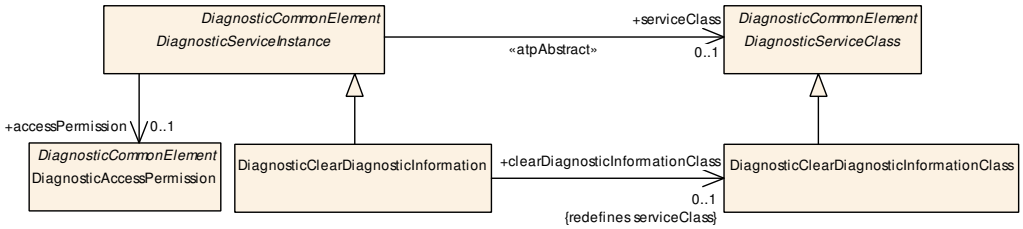

Figure 4.36: Modeling of diagnostic service `ClearDiagnosticInformation` (0x14)

Class	**DiagnosticClearDiagnosticInformation**			
Note	This represents an instance of the "Clear Diagnostic Information" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticClearDiagnosticInformations			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
clearDiagnostic Information Class	DiagnosticClear DiagnosticInformation Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticClearDiagnostic Information in the given context. **Tags:** atp.Splitkey=clearDiagnosticInformationClass

Table 4.108: DiagnosticClearDiagnosticInformation

Class	**DiagnosticClearDiagnosticInformationClass**			
Note	This meta-class contains attributes shared by all instances of the "Clear Diagnostic Information" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticClearDiagnosticInformations			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.109: DiagnosticClearDiagnosticInformationClass

[TPS_DEXT_01022] `ClearDiagnosticInformation` does not define any sub-functions ⌈The diagnostic service `ClearDiagnosticInformation` does not define any sub-functions, therefore the value of `DiagnosticClearDiagnosticInformation.category` does not need to be constrained.⌋

4.3.8.15 Memory Services

This chapter describes the modeling of diagnostic services for memory access (0x23, 0x3D, 0x34-0x37). The purpose of these services is to access memory on the diagnostic stack on request of the tester.

The service description for accessing memory for diagnostic purposes is modeled by the abstract meta-class `DiagnosticMemoryByAddress`. It is supposed to provide all model properties relevant for the memory access.

The description of memory access, to some extent, requires a formal description of the memory segments to take into account. For this purpose the meta-class `DiagnosticMemoryIdentifier` has been introduced and referenced by `DiagnosticMemoryAddressableRangeAccess` in the role `memoryRange`.

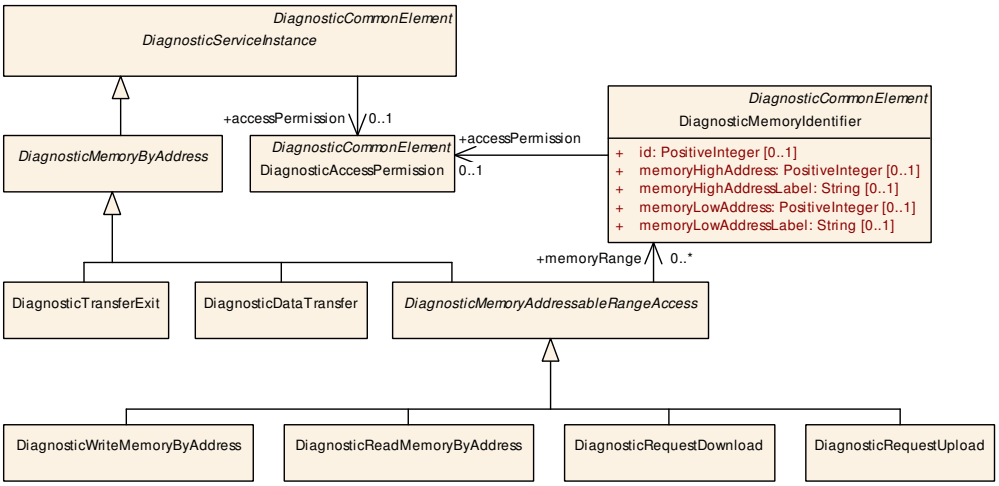

Figure 4.37: Modeling of diagnostic services `Memory` (0x23, 0x3D, 0x34-0x37)

The intent of this modeling was not to provide a generic memory model but to allow for the specification of memory properties just as far as diagnostics is concerned.

The aggregation of DiagnosticMemoryIdentifier at DiagnosticMemoryByAddress may or may not be relevant for an OEM. However, there is certainly a use case for adding this information to a DiagnosticExtract that goes back from a tier-1 supplier to an OEM as sort of documentation of the diagnostic configuration.

As DiagnosticMemoryByAddress represents a generic base class for all kinds of diagnostic memory access, it is also necessary to model the particular sub-classes that address specific use cases for diagnostic memory access.

These sub-classes are conceptually on the same level as other sub-classes of DiagnosticServiceInstance.

In other words, the case of memory access deviates from the modeling of other diagnostic services such that there is one further abstract base class involved.

[constr_1333] Existence of attributes of meta-class DiagnosticMemoryIdentifier

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈If a DiagnosticMemoryIdentifier is referenced in the role memoryRange by a DiagnosticRequestDownload or a DiagnosticRequestUpload, then the attributes

- memoryLowAddress
- memoryLowAddressLabel
- memoryHighAddress
- memoryHighAddressLabel
- accessPermission

shall **not** exist for the referenced DiagnosticMemoryIdentifier.⌋

[constr_1411] Existence of attribute DiagnosticMemoryIdentifier.memoryHighAddressLabel vs. DiagnosticMemoryIdentifier.memoryHighAddress

Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈At most **one** of the attributes in the following list shall exist:

- DiagnosticMemoryIdentifier.memoryHighAddressLabel
- DiagnosticMemoryIdentifier.memoryHighAddress

⌋

[constr_1412] Existence of `DiagnosticMemoryIdentifier.memoryLowAddressLabel` vs. `DiagnosticMemoryIdentifier.memoryLowAddress`

Imposition time: CP: `IT_DiagDes`, AP: `IT_DiagDes`

⌈At most **one** of the attributes in the following list shall exist:

- `DiagnosticMemoryIdentifier.memoryLowAddressLabel`
- `DiagnosticMemoryIdentifier.memoryLowAddress`

⌋

Please note that it does not make sense to describe a memory address in this context **both** numerically **and** symbolically. If the address is described at all (see [constr_1333]) then it shall be done **either** symbolically or numerically. This is the motivation of the existence of [constr_1411] and [constr_1412].

[TPS_DEXT_01023] `WriteMemoryByAddress` does not define any sub-functions
⌈The diagnostic service `WriteMemoryByAddress` does not define any sub-functions, therefore the value of `DiagnosticWriteMemoryByAddress.category` does not need to be constrained.⌋

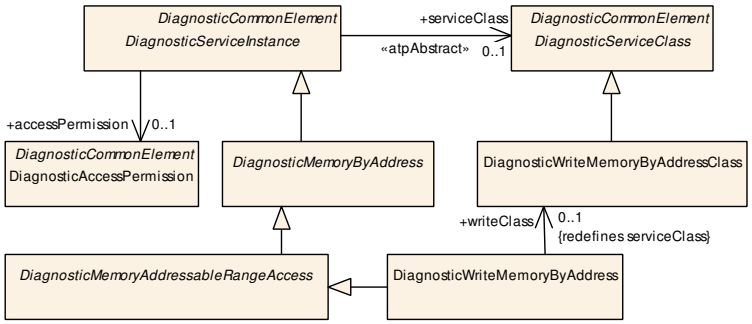

Figure 4.38: Modeling of diagnostic service `WriteMemoryByAddress` (0x3D)

Class	***DiagnosticMemoryByAddress*** (abstract)			
Note	This represents an abstract base class for diagnostic services that deal with accessing memory by address.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticDataTransfer, *DiagnosticMemoryAddressableRangeAccess*, DiagnosticTransferExit			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.110: DiagnosticMemoryByAddress

Class	***DiagnosticMemoryAddressableRangeAccess*** (abstract)
Note	This abstract base class
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemoryByAddress*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*

△

Class	**DiagnosticMemoryAddressableRangeAccess** (abstract)			
Subclasses	DiagnosticReadMemoryByAddress, DiagnosticRequestDownload, DiagnosticRequestUpload, DiagnosticWriteMemoryByAddress			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
memoryRange	DiagnosticMemoryIdentifier	*	ref	This represents the formal description of the memory segment to which the DiagnosticMemoryByAddress applies. **Tags:** atp.Splitkey=memoryRange

Table 4.111: DiagnosticMemoryAddressableRangeAccess

Class	**DiagnosticMemoryIdentifier**			
Note	This meta-class represents the ability to define memory properties from the diagnostics point of view. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
accessPermission	DiagnosticAccessPermission	0..1	ref	This represents that access permission defined for the specific DiagnosticMemoryIdentifier. **Tags:** atp.Splitkey=accessPermission
id	PositiveInteger	0..1	attr	This represents the identification of the memory segment.
memoryHighAddress	PositiveInteger	0..1	attr	This represents the upper bound for addresses of the memory segment.
memoryHighAddressLabel	String	0..1	attr	This represents a symbolic label for the upper bound for addresses of the memory segment.
memoryLowAddress	PositiveInteger	0..1	attr	This represents the lower bound for addresses of the memory segment.
memoryLowAddressLabel	String	0..1	attr	This represents a symbolic label for the lower bound for addresses of the memory segment.

Table 4.112: DiagnosticMemoryIdentifier

Class	**DiagnosticWriteMemoryByAddress**			
Note	This represents an instance of the "Write Memory by Address" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemoryAddressableRangeAccess*, *DiagnosticMemoryByAddress*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
writeClass	DiagnosticWriteMemoryByAddressClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticWritememoryBy Address in the given context. **Tags:** atp.Splitkey=writeClass

Table 4.113: DiagnosticWriteMemoryByAddress

Class	DiagnosticWriteMemoryByAddressClass			
Note	This meta-class contains attributes shared by all instances of the "Write Memory by Address" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.114: DiagnosticWriteMemoryByAddressClass

Class	DiagnosticReadMemoryByAddress			
Note	This represents an instance of the "Read Memory by Address" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemoryAddressableRangeAccess*, *DiagnosticMemoryByAddress*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
readClass	DiagnosticReadMemoryByAddressClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticReadMemoryByAddresst in the given context. **Tags:** atp.Splitkey=readClass

Table 4.115: DiagnosticReadMemoryByAddress

Class	DiagnosticReadMemoryByAddressClass			
Note	This meta-class contains attributes shared by all instances of the "Read Memory by Address" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.116: DiagnosticReadMemoryByAddressClass

[TPS_DEXT_01024] `ReadMemoryByAddress` does not define any sub-functions
⌈The diagnostic service `ReadMemoryByAddress` does not define any sub-functions, therefore the value of `DiagnosticReadMemoryByAddress.category` does not need to be constrained.⌋

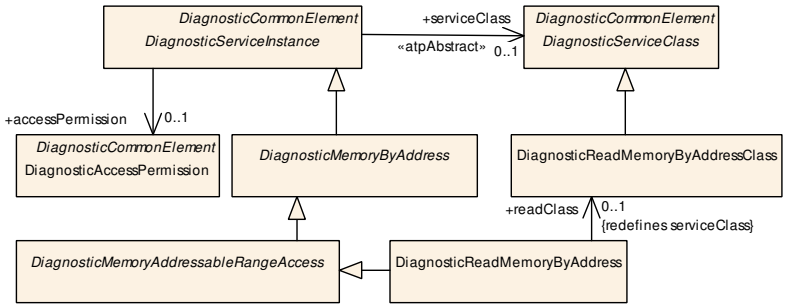

Figure 4.39: Modeling of diagnostic service `ReadMemoryByAddress` (0x23)

Class	**DiagnosticTransferExit**			
Note	This represents an instance of the "Transfer Exit" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemoryByAddress*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
transferExit Class	DiagnosticTransferExit Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticTransferExit in the given context. **Tags:** atp.Splitkey=transferExitClass

Table 4.117: DiagnosticTransferExit

Class	**DiagnosticTransferExitClass**			
Note	This meta-class contains attributes shared by all instances of the "Transfer Exit" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.118: DiagnosticTransferExitClass

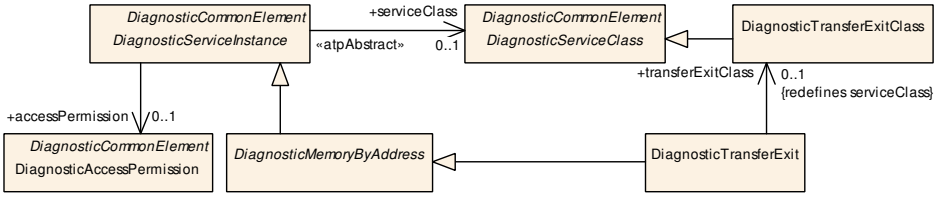

Figure 4.40: Modeling of diagnostic service `TransferExit` (0x37)

[TPS_DEXT_01025] `TransferExit` does not define any sub-functions ⌈The diagnostic service `TransferExit` does not define any sub-functions, therefore the value of `DiagnosticTransferExit.category` does not need to be constrained.⌋

Class	DiagnosticDataTransfer			
Note	This represents an instance of the "Data Transfer" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemoryByAddress*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataTransferClass	DiagnosticDataTransferClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticDataTransfer in the given context. **Tags:** atp.Splitkey=dataTransferClass

Table 4.119: DiagnosticDataTransfer

Class	DiagnosticDataTransferClass			
Note	This meta-class contains attributes shared by all instances of the "Data Transfer" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.120: DiagnosticDataTransferClass

[TPS_DEXT_01026] `DataTransfer` does not define any sub-functions ⌈The diagnostic service `DataTransfer` does not define any sub-functions, therefore the value of `DiagnosticDataTransfer.category` does not need to be constrained.⌋

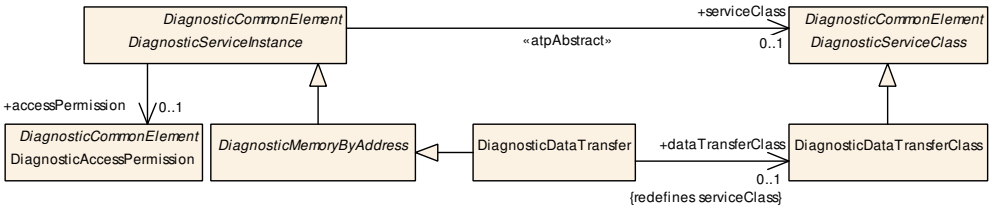

Figure 4.41: Modeling of diagnostic service `DataTransfer` (0x36)

Class	DiagnosticRequestDownload			
Note	This represents an instance of the "Request Download" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemoryAddressableRangeAccess*, *DiagnosticMemoryByAddress*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

△

Class	**DiagnosticRequestDownload**			
request DownloadClass	DiagnosticRequest DownloadClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestDownload in the given context. **Tags:** atp.Splitkey=requestDownloadClass

Table 4.121: DiagnosticRequestDownload

Class	**DiagnosticRequestDownloadClass**			
Note	This meta-class contains attributes shared by all instances of the "Request Download" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.122: DiagnosticRequestDownloadClass

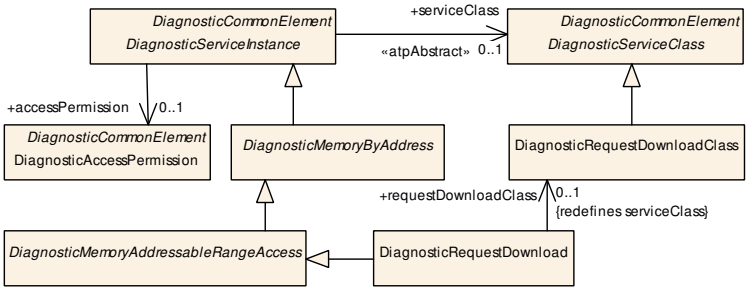

Figure 4.42: Modeling of diagnostic service `RequestDownload` (0x34)

[TPS_DEXT_01027] `RequestDownload` does not define any sub-functions ⌈The diagnostic service `RequestDownload` does not define any sub-functions, therefore the value of `DiagnosticRequestDownload.category` does not need to be constrained.⌋

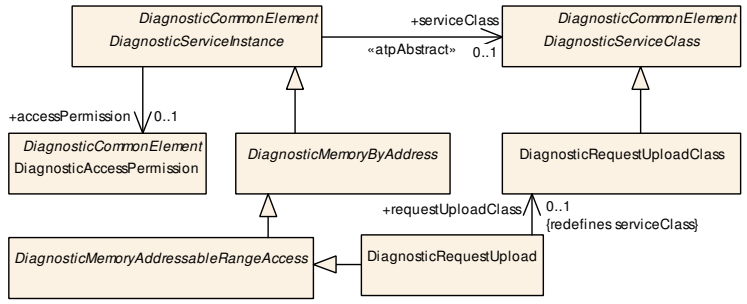

Figure 4.43: Modeling of diagnostic service `RequestUpload` (0x35)

Class	DiagnosticRequestUpload			
Note	This represents an instance of the "Request Upload" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemoryAddressableRangeAccess*, *DiagnosticMemoryByAddress*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
requestUpload Class	DiagnosticRequest UploadClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestUpload in the given context. **Tags:** atp.Splitkey=requestUploadClass

Table 4.123: DiagnosticRequestUpload

Class	DiagnosticRequestUploadClass			
Note	This meta-class contains attributes shared by all instances of the "Request Upload" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.124: DiagnosticRequestUploadClass

[TPS_DEXT_01028] `RequestUpload` does not define any sub-functions ⌈The diagnostic service `RequestUpload` does not define any sub-functions, therefore the value of `DiagnosticRequestUpload.category` does not need to be constrained.⌋

4.3.8.16 RequestFileTransfer

This chapter describes the modeling of diagnostic services `RequestFileTransfer` (0x38). The purpose of the service is the triggering of the transfer of a *file* from or to the AUTOSAR diagnostic stack.

Please note that there is nothing to configure for `DiagnosticRequestFileTransfer` beyond its mere existence.

[TPS_DEXT_01090] Diagnostic service `RequestFileTransfer` does not define any sub-functions ⌈The diagnostic service `RequestFileTransfer` does not define any sub-functions. Therefore, the usage of the attribute `category` is not constrained for meta-class `DiagnosticRequestFileTransfer`.⌋

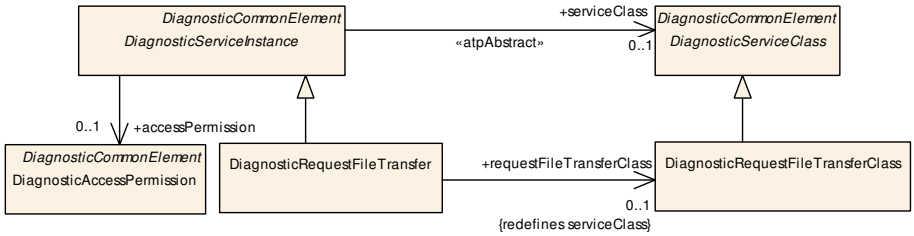

Figure 4.44: Modeling of diagnostic service `RequestFileTransfer` (0x38)

Class	**DiagnosticRequestFileTransfer**			
Note	This diagnostic service instance implements the UDS service 0x38. **Tags:** atp.recommendedPackage=DiagnosticRequestFileTransfers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
requestFile TransferClass	DiagnosticRequestFile TransferClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestFile Transfer in the given context. **Tags:** atp.Splitkey=requestFileTransferClass

Table 4.125: DiagnosticRequestFileTransfer

Class	**DiagnosticRequestFileTransferClass**			
Note	This meta-class contains attributes shared by all instances of the "Request File transfer" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRequestFileTransfers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.126: DiagnosticRequestFileTransferClass

4.3.9 OBD Diagnostic Services supported by AUTOSAR

Support for diagnostic services for on-board diagnostics (OBD) (see [19, ISO 15031-5]) requires the definition of further meta-classes similar to e.g. DiagnosticData-Identifier or DiagnosticRoutine.

[TPS_DEXT_01092] Semantics of `DiagnosticParameterIdentifier` [The meta-class DiagnosticParameterIdentifier is used to reflect the concept of the so-called Parameter Identifiers (PID) in the diagnostic extract. A Diagnostic-ParameterIdentifier defines the following properties:

- Each DiagnosticParameterIdentifier gets a numerical ID carried in the attribute DiagnosticParameterIdentifier.id.
- DiagnosticParameterIdentifier may also contain a so-called *Support Info Byte*, modeled as DiagnosticSupportInfoByte.
- The definition of a DiagnosticParameterIdentifier also contains a list of data associated with the PID. This list is modeled as an aggregation of DiagnosticParameter.

⌋

As already mentioned in [TPS_DEXT_01092], the DiagnosticParameter-Identifier shall not be confused with the DiagnosticParameter. The latter is used "inside" the definition of DiagnosticParameterIdentifier, but also in DiagnosticDataIdentifier or DiagnosticRoutine, to define one (out of potentially many) piece of information held in the scope of a DiagnosticParameter-Identifier.

[constr_1447] Restrictions for the value of DiagnosticParameterIdentifier.id

Imposition time: CP: IT_DiagDes

⌈The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of DiagnosticParameterIdentifier.id.⌋

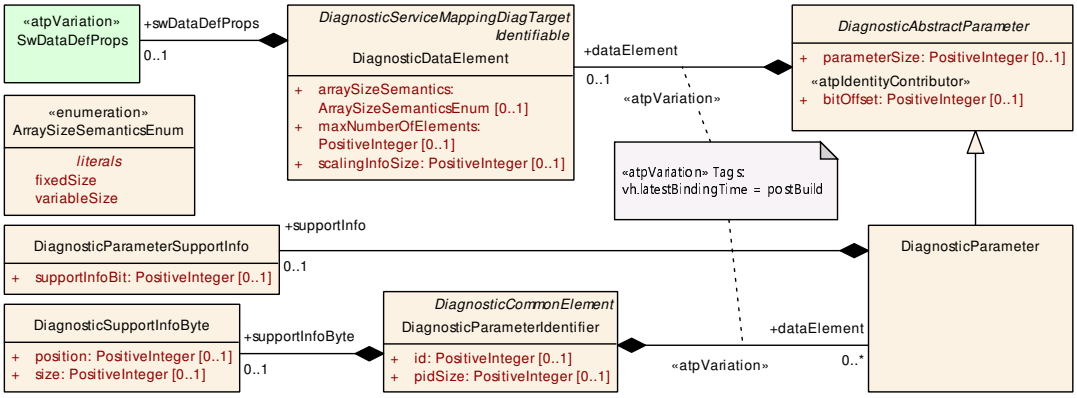

Figure 4.45: Modeling of the DiagnosticParameterIdentifier

[constr_1448] Interval of DiagnosticParameterIdentifier.id

Imposition time: CP: IT_DiagDes

⌈The allowed interval for values of DiagnosticParameterIdentifier.id shall not exceed [0..255].⌋

[constr_10370] Restriction regarding the role DiagnosticParameterIdentifier.dataElement

Imposition time: CP: IT_DiagDes

⌈A DiagnosticParameter that is aggregated by a DiagnosticParameterIdentifier in the role dataElement shall not aggregate in the role ident a DiagnosticParameterIdent that in turn aggregates in the role subElement a DiagnosticParameterElement.⌋

The rationale for [constr_10370] is that the usage of DiagnosticParameter in the context of the DiagnosticParameterIdentifier is typically limited to a few primitive values and therefore does not have a need for nested structured DiagnosticParameter.

[constr_1449] PID shall only carry a fixed-length collection of data

Imposition time: CP: IT_DiagDes

⌈The value of DiagnosticParameterIdentifier.dataElement.dataElement.arraySizeSemantics shall not be set to variableSize.⌋

[constr_1750] Existence of attribute DiagnosticParameterIdentifier.pidSize

Imposition time: CP: IT_DiagDes

⌈Attribute DiagnosticParameterIdentifier.pidSize is only relevant if a gap exists at the end of the DiagnosticParameterIdentifier. If this gap does not exist, the size of the DiagnosticParameterIdentifier can be computed.⌋

Class	**DiagnosticParameterIdentifier**			
Note	This meta-class represents the ability to model a diagnostic parameter identifier (PID) for the purpose of executing on-board diagnostics (OBD). **Tags:** atp.recommendedPackage=DiagnosticParameterIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataElement	DiagnosticParameter	*	aggr	This represents the data carried by the Diagnostic ParameterIdentifier. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataElement, dataElement.bitOffset, data Element.ident.shortName, dataElement.variation Point.shortLabel vh.latestBindingTime=postBuild
id	PositiveInteger	0..1	attr	This is the numerical identifier used to identify the DiagnosticParameterIdentifier in the scope of diagnostic workflow (see SAE J1979-DA).
pidSize	PositiveInteger	0..1	attr	The size of the entire PID can be greater than the sum of the data elements because padding might be applied. Unit: byte.

▽

△

Class	DiagnosticParameterIdentifier			
supportInfoByte	DiagnosticSupportInfoByte	0..1	aggr	This represents the supported information associated with the DiagnosticParameterIdentifier. **Tags:** atp.Splitkey=supportInfoByte

Table 4.127: DiagnosticParameterIdentifier

[constr_1819] Existence of attribute DiagnosticParameterIdentifier.id

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticParameterIdentifier, attribute id shall exist.⌋

[constr_10545] Existence of DiagnosticParameterIdentifier.dataElement

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticParameterIdentifier, at least one aggregation of DiagnosticParameter in the role DiagnosticParameterIdentifier.dataElement shall exist.⌋

Class	DiagnosticParameterSupportInfo			
Note	This represents a way to define which bit of the supportInfo is representing this part of the PID			
Base	*ARObject*			
Aggregated by	DiagnosticParameter.supportInfo			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
supportInfoBit	PositiveInteger	0..1	attr	defines the bit in the SupportInfo byte, which represents the PID DataElement pidSize / position / size. Unit: byte.

Table 4.128: DiagnosticParameterSupportInfo

Class	DiagnosticSupportInfoByte			
Note	This meta-class defines the support information (typically byte A) to declare the usability of the Data Elements within the so-called packeted PIDs (e.g. PID$68).			
Base	*ARObject*			
Aggregated by	DiagnosticDataIdentifier.supportInfoByte, DiagnosticParameterIdentifier.supportInfoByte			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
position	PositiveInteger	0..1	attr	This represents the position of the supportInfo in the PID. Unit: byte.
size	PositiveInteger	0..1	attr	This represents the size of the supportInfo within the PID. Unit: byte.

Table 4.129: DiagnosticSupportInfoByte

4.3.9.1 OBD Mode 0x01 (RequestCurrentPowertrainDiagnosticData)

The service RequestCurrentPowertrainDiagnosticData is modeled in Figure 4.46.

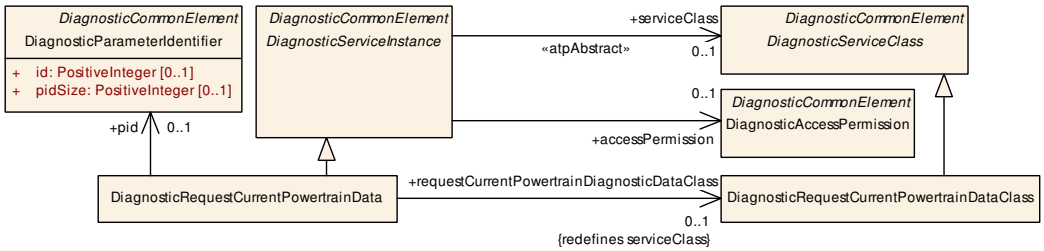

Figure 4.46: Modeling of diagnostic service for OBD Mode 0x01

[TPS_DEXT_01125] Support for diagnostic service `RequestCurrentPowertrainDiagnosticData` ⌈The modeling support for the OBD diagnostic service `RequestCurrentPowertrainDiagnosticData` utilizes the new meta-class `DiagnosticParameterIdentifier` such that the meta-class `DiagnosticRequestCurrentPowertrainData` maintains a reference to `DiagnosticParameterIdentifier` in the role `pid`.⌋

Class	**DiagnosticRequestCurrentPowertrainData**			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x01 service. **Tags:** atp.recommendedPackage=DiagnosticRequestCurrentPowertrainDatas			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
pid	DiagnosticParameter Identifier	0..1	ref	This represents the PID associated with this instance of the OBD mode 0x01 service. **Tags:** atp.Splitkey=pid
requestCurrent Powertrain DiagnosticData Class	DiagnosticRequest CurrentPowertrainData Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestCurrent PowertrainData in the given context. **Tags:** atp.Splitkey=requestCurrentPowertrainDiagnostic DataClass

Table 4.130: DiagnosticRequestCurrentPowertrainData

[constr_1820] Existence of reference in the role `DiagnosticRequestCurrentPowertrainData.pid`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticRequestCurrentPowertrainData`, the reference to meta-class `DiagnosticParameterIdentifier` in the role `pid` shall exist.⌋

Class	**DiagnosticRequestCurrentPowertrainDataClass**
Note	This meta-class represents the ability to define common properties for all instances of the "Request current Powertrain Data" OBD diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRequestCurrentPowertrainDatas

▽

△

Class	DiagnosticRequestCurrentPowertrainDataClass			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 4.131: DiagnosticRequestCurrentPowertrainDataClass

4.3.9.2 OBD Mode 0x02 (RequestPowertrainFreezeFrameData)

The service `RequestPowertrainFreezeFrameData` is modeled in Figure 4.47.

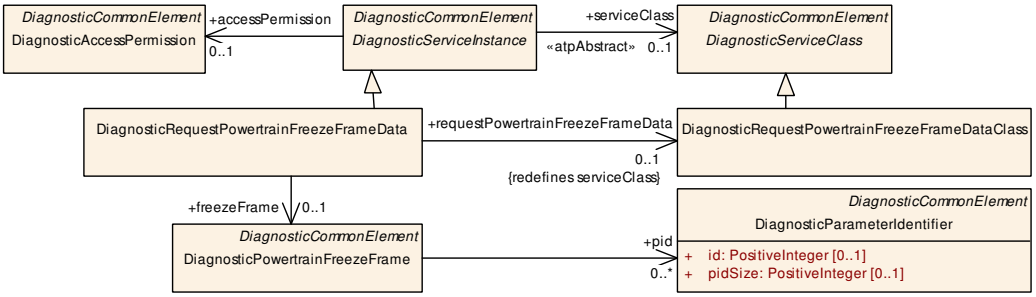

Figure 4.47: Modeling of diagnostic service for OBD Mode 0x02

[TPS_DEXT_01126] Support of OBD service `RequestPowertrainFreezeFrameData` ⌈The modeling support for the ODB diagnostic service `RequestPowertrainFreezeFrameData` utilizes the new meta-class `DiagnosticParameterIdentifier` such that the meta-class `DiagnosticRequestCurrentPowertrainData` maintains a reference to `DiagnosticParameterIdentifier` in the role `pid`.⌋

Class	DiagnosticRequestPowertrainFreezeFrameData			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x02 service. **Tags:** atp.recommendedPackage=DiagnosticPowertrainFreezeFrames			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
freezeFrame	DiagnosticPowertrain FreezeFrame	0..1	ref	This represents the associated freeze-frame. **Tags:** atp.Splitkey=freezeFrame

▽

△

Class	DiagnosticRequestPowertrainFreezeFrameData			
request PowertrainFreezeFrameData	DiagnosticRequestPowertrainFreezeFrameDataClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestPowertrainFreezeFrameData in the given context. **Tags:** atp.Splitkey=requestPowertrainFreezeFrameData

Table 4.132: DiagnosticRequestPowertrainFreezeFrameData

[constr_1821] Existence of reference DiagnosticRequestPowertrainFreeze-FrameData.freezeFrame

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticRequestPowertrainFreezeFrameData, the reference to DiagnosticParameterIdentifier in the role freezeFrame shall exist.⌋

Class	DiagnosticRequestPowertrainFreezeFrameDataClass			
Note	This meta-class represents the ability to define common properties for all instances of the "Request Powertrain Freeze Frame Data" OBD diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticPowertrainFreezeFrames			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.133: DiagnosticRequestPowertrainFreezeFrameDataClass

Class	DiagnosticPowertrainFreezeFrame			
Note	This meta-class represents a powertrain-related freeze-frame. In theory, this meta-class would need an additional id attribute. However, legal regulations requires only a single value for this attribute anyway. **Tags:** atp.recommendedPackage=DiagnosticPowertrainFreezeFrames			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
pid	DiagnosticParameterIdentifier	*	ref	This represents the PID associated with this instance of the OBD mode 0x02 service. **Tags:** atp.Splitkey=pid

Table 4.134: DiagnosticPowertrainFreezeFrame

4.3.9.3 OBD Mode 0x03 / 0x07 (RequestEmissionRelatedDiagnosticTroubleCodes)

[TPS_DEXT_01127] Semantics of meta-class DiagnosticRequestEmissionRelatedDTC ⌈Two very similar OBD services, subsumed as RequestEmissionRelatedDiagnosticTroubleCodes are supported by means of the meta-class DiagnosticRequestEmissionRelatedDTC.⌋

The modeling is sketched in Figure 4.48.

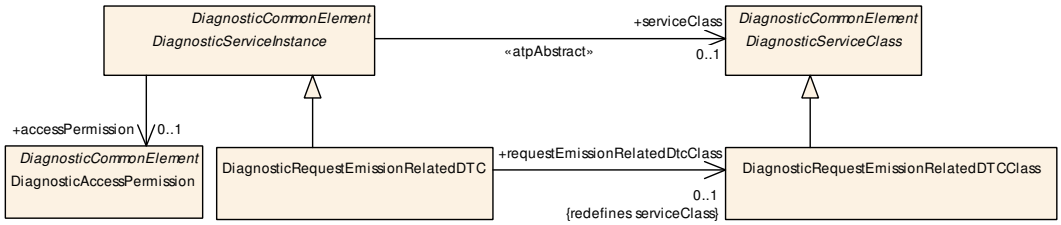

Figure 4.48: Modeling of diagnostic service for OBD Modes 0x03, 0x07

Class	**DiagnosticRequestEmissionRelatedDTC**			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x03/0x07 service. **Tags:** atp.recommendedPackage=DiagnosticRequestEmissionRelatedDTCs			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
request Emission RelatedDtc Class	DiagnosticRequest EmissionRelated DTCClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestEmission RelatedDTC in the given context. **Tags:** atp.Splitkey=requestEmissionRelatedDtcClass

Table 4.135: DiagnosticRequestEmissionRelatedDTC

Class	**DiagnosticRequestEmissionRelatedDTCClass**			
Note	This meta-class represents the ability to define common properties for all instances of the "Request Emission Related DTC" OBD diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRequestEmissionRelatedDTCs			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.136: DiagnosticRequestEmissionRelatedDTCClass

4.3.9.4 OBD Mode 0x04 (ClearResetEmissionRelatedDiagnosticInformation)

[TPS_DEXT_01128] Semantics of meta-class DiagnosticClearReset-EmissionRelatedInfo ⌈The OBD diagnostic service ClearResetEmissionRe-latedDiagnosticInformation is supported by means of the meta-class DiagnosticClearResetEmissionRelatedInfo.⌋

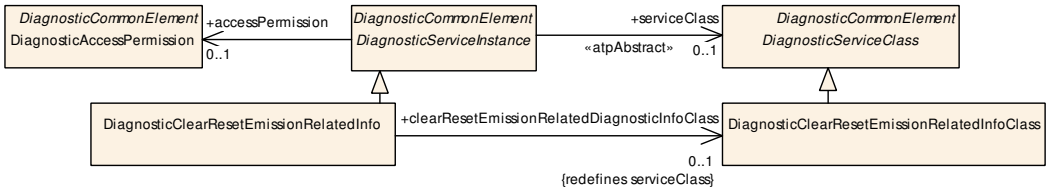

Figure 4.49: Modeling of diagnostic service for OBD Mode 0x04

The modeling is sketched in Figure 4.49.

Class	**DiagnosticClearResetEmissionRelatedInfo**			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x04 service. **Tags:** atp.recommendedPackage=DiagnosticClearResetEmissionRelatedInfos			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
clearReset Emission Related DiagnosticInfo Class	DiagnosticClearReset EmissionRelatedInfo Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticClearReste EmissionRelatedInfo in the given context. **Tags:** atp.Splitkey=clearResetEmissionRelatedDiagnostic InfoClass

Table 4.137: DiagnosticClearResetEmissionRelatedInfo

Class	**DiagnosticClearResetEmissionRelatedInfoClass**			
Note	This meta-class represents the ability to define common properties for all instances of the "Clear Reset Emission Related Data" OBD diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticClearResetEmissionRelatedInfos			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.138: DiagnosticClearResetEmissionRelatedInfoClass

4.3.9.5 OBD Mode 0x06 (RequestOnBoardMonitoringTestResults)

[TPS_DEXT_01129] Support for OBD diagnostic service `RequestOnBoardMonitoringTestResults` ⌈The OBD diagnostic service `RequestOnBoardMonitoringTestResults` is supported by manes of meta-class `DiagnosticRequestOnBoardMonitoringTestResults` that refers to the representation of the test result modeled as meta-class `DiagnosticTestResult`⌋

The modeling is sketched in Figure 4.50.

In general, the mode 0x06 supports the querying of supported test identifiers from a given server. However, this functionality is not supported in AUTOSAR, hence the existence of [constr_1462].

[constr_1462] Restrictions for the value of `DiagnosticTestResult.testIdentifier.id`

Imposition time: CP: `IT_DiagDes`

⌈The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of `DiagnosticTestResult.testIdentifier.id`.⌋

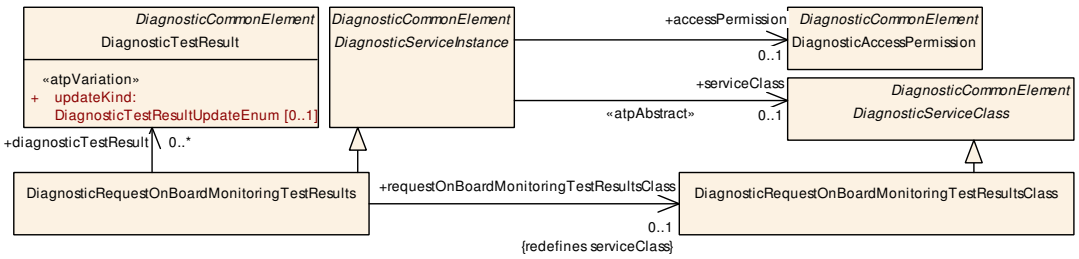

Figure 4.50: Modeling of diagnostic service for OBD Mode 0x06

[TPS_DEXT_01141] Definition of a collection of test results ⌈It is possible to assign a collection of test results to the mode 0x06.

For this purpose, `DiagnosticRequestOnBoardMonitoringTestResults` shall refer to multiple `DiagnosticTestResult`s in the role `diagnosticTestResult` that in turn refers to the identical `DiagnosticMeasurementIdentifier` in the role `monitoredIdentifier` (taken as the identification of related test results in downstream processing of a diagnostic extract).⌋

Class	**DiagnosticRequestOnBoardMonitoringTestResults**			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x06 service. **Tags:** atp.recommendedPackage=DiagnosticRequestOnBoardMonitoringTestResultss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**DiagnosticRequestOnBoardMonitoringTestResults**			
diagnosticTest Result	DiagnosticTestResult	*	ref	This reference identifies the applicable collection of test identifiers for setting up a request message for mode 0x06. **Tags:** atp.Splitkey=diagnosticTestResult
requestOn Board MonitoringTest ResultsClass	DiagnosticRequestOn BoardMonitoringTest ResultsClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestOnBoard MonitoringTestResults in the given context. **Tags:** atp.Splitkey=requestOnBoardMonitoringTest ResultsClass

Table 4.139: DiagnosticRequestOnBoardMonitoringTestResults

Class	**DiagnosticRequestOnBoardMonitoringTestResultsClass**			
Note	This meta-class represents the ability to define common properties for all instances of the "Request On-Board Monitoring Test Results" OBD diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRequestOnBoardMonitoringTestResultss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table 4.140: DiagnosticRequestOnBoardMonitoringTestResultsClass

4.3.9.6 OBD Mode 0x08 (RequestControlOfOnBoardDevice)

[TPS_DEXT_01130] Support of OBD diagnostic service `RequestControlOfOnBoardDevice` ⌈The OBD diagnostic service `RequestControlOfOnBoardDevice` is supported by means of meta-class `DiagnosticRequestControlOfOnBoardDevice` that in turn refers to a `DiagnosticTestRoutineIdentifier` in the role `testId`⌋

The modeling is sketched in Figure 4.51.

In general, the mode 0x06 supports the querying of supported test routine identifiers from a given server. However, this functionality is not supported in AUTOSAR, hence the existence of [constr_1461].

[constr_1461] Restrictions for the value of `DiagnosticTestRoutineIdentifier.id`

Imposition time: CP: `IT_DiagDes`

⌈The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of `DiagnosticTestRoutineIdentifier.id`.⌋

Class	DiagnosticRequestControlOfOnBoardDevice			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x08 service. **Tags:** atp.recommendedPackage=DiagnosticRequestControlOfOnBoardDevices			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
requestControlOfOnBoardDeviceClass	DiagnosticRequestControlOfOnBoardDeviceClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestControlOfOnBoardDevice in the given context. **Tags:** atp.Splitkey=requestControlOfOnBoardDeviceClass
testId	DiagnosticTestRoutineIdentifier	0..1	ref	This represents the test Id for the mode 0x08. **Tags:** atp.Splitkey=testId

Table 4.141: DiagnosticRequestControlOfOnBoardDevice

[constr_1822] Existence of reference DiagnosticRequestControlOfOnBoard-Device.testId

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticRequestControlOfOnBoardDevice, the reference to DiagnosticParameterIdentifier in the role testId shall exist.⌋

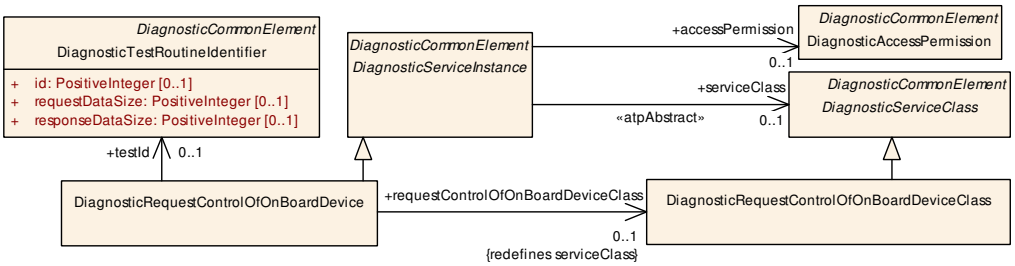

Figure 4.51: Modeling of diagnostic service for OBD Mode 0x08

Class	DiagnosticRequestControlOfOnBoardDeviceClass			
Note	This meta-class represents the ability to define common properties for all instances of the "Request Control Of On-Board Device" OBD diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRequestControlOfOnBoardDevices			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.142: DiagnosticRequestControlOfOnBoardDeviceClass

Class	DiagnosticTestRoutineIdentifier			
Note	This represents the test id of the DiagnosticTestIdentifier. **Tags:** atp.recommendedPackage=DiagnosticTestRoutineIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
id	PositiveInteger	0..1	attr	This represents the numerical id of the DiagnosticTest Identifier (see SAE J1979-DA).
requestData Size	PositiveInteger	0..1	attr	This represents the specified data size for the request message. Unit: byte.
responseData Size	PositiveInteger	0..1	attr	This represents the specified data size for the response message. Unit:byte.

Table 4.143: DiagnosticTestRoutineIdentifier

[constr_1823] Existence of attribute `DiagnosticTestRoutineIdentifier.id`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticTestRoutineIdentifier`, attribute `id` shall exist.⌋

[constr_1824] Existence of attribute `DiagnosticTestRoutineIdentifier.requestDataSize`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticTestRoutineIdentifier`, attribute `requestDataSize` shall exist.⌋

[constr_1825] Existence of attribute `DiagnosticTestRoutineIdentifier.responseDataSize`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticTestRoutineIdentifier`, attribute `responseDataSize` shall exist.⌋

4.3.9.7 OBD Mode 0x09 (RequestVehicleInformation)

[TPS_DEXT_01131] Support for OBD diagnostic service `RequestVehicleInformation` ⌈The `OBD` diagnostic service `RequestVehicleInformation` is supported by means of meta-class `DiagnosticRequestVehicleInfo` that in turn references a `DiagnosticInfoType` in the role `infoType`.⌋

[constr_1460] Restrictions for the value of `DiagnosticInfoType.id`

Imposition time: CP: `IT_DiagDes`

⌈The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of `DiagnosticInfoType.id`.⌋

The modeling is sketched in Figure 4.52.

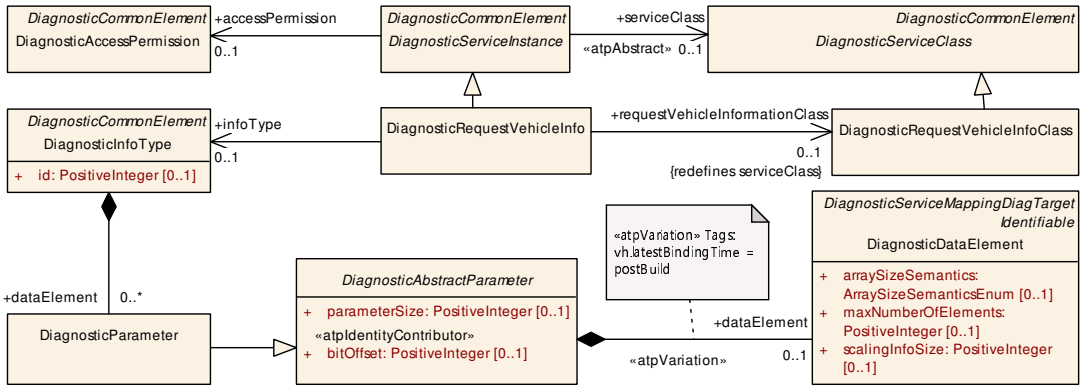

Figure 4.52: Modeling of diagnostic service for OBD Mode 0x09

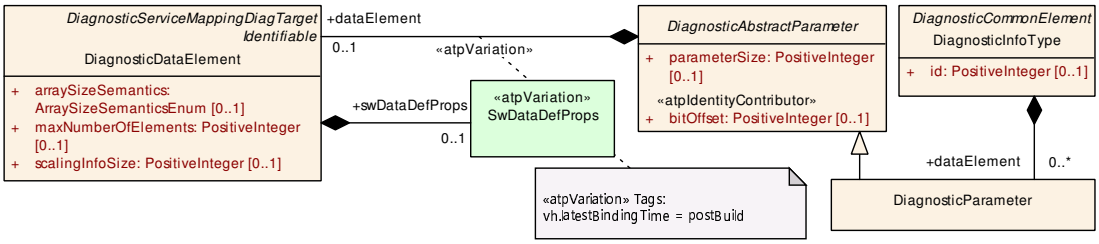

Figure 4.53: Modeling of DiagnosticInfoType

[constr_1749] Existence of DiagnosticInfoType.dataElement

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticInfoType, at least one aggregation of DiagnosticParameter in the role dataElement shall exist.⌋

Class	**DiagnosticRequestVehicleInfo**			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x09 service. **Tags:** atp.recommendedPackage=DiagnosticRequestVehicleInfos			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
infoType	DiagnosticInfoType	0..1	ref	This represents the info type associated with the mode 0x09 service. **Tags:** atp.Splitkey=infoType
requestVehicle Information Class	DiagnosticRequest VehicleInfoClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequesVehicleInfo in the given context. **Tags:** atp.Splitkey=requestVehicleInformationClass

Table 4.144: DiagnosticRequestVehicleInfo

[constr_1826] Existence of reference `DiagnosticRequestVehicleInfo.infoType`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticRequestVehicleInfo`, the reference to `DiagnosticParameterIdentifier` in the role `infoType` shall exist.⌋

Class	**DiagnosticRequestVehicleInfoClass**			
Note	This meta-class represents the ability to define common properties for all instances of the "Request Vehicle Info" OBD diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRequestVehicleInfos			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.145: DiagnosticRequestVehicleInfoClass

Class	**DiagnosticInfoType**			
Note	This meta-class represents the ability to model an OBD info type. **Tags:** atp.recommendedPackage=DiagnosticInfoTypes			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataElement	DiagnosticParameter	*	aggr	This represents the data associated with the enclosing DiagnosticInfoType. **Tags:** atp.Splitkey=dataElement, dataElement.bitOffset, dataElement.ident.shortName
id	PositiveInteger	0..1	attr	This attribute represents the value of InfoType (see SAE J1979-DA).

Table 4.146: DiagnosticInfoType

[constr_1827] Existence of attribute `DiagnosticInfoType.id`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticInfoType`, attribute `id` shall exist.⌋

4.3.9.8 OBD Mode 0x0A (RequestEmissionRelatedDiagnosticTroubleCodesPermanentStatus)

[TPS_DEXT_01132] Support for OBD diagnostic service `RequestEmissionRelatedDiagnosticTroubleCodesPermanentStatus` ⌈The `OBD` diagnostic service `RequestEmissionRelatedDiagnosticTroubleCodesPermanentStatus` is supported by means of meta-class `DiagnosticRequestEmissionRelatedDTCPermanentStatus`.⌋

The modeling is sketched in Figure 4.54.

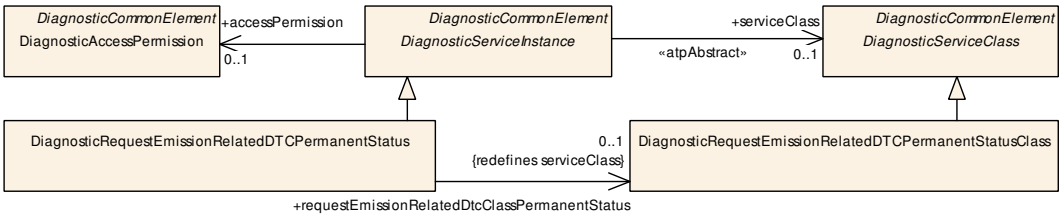

Figure 4.54: Modeling of diagnostic service for OBD Mode 0x0A

Class	**DiagnosticRequestEmissionRelatedDTCPermanentStatus**			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x0A service. **Tags:** atp.recommendedPackage=DiagnosticRequestEmissionRelatedDTCPermanentStatuss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
request Emission RelatedDtc Class Permanent Status	DiagnosticRequest EmissionRelated DTCPermanentStatus Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestEmission RelatedDTCPermanentStatus in the given context. **Tags:** atp.Splitkey=requestEmissionRelatedDtcClass PermanentStatus

Table 4.147: DiagnosticRequestEmissionRelatedDTCPermanentStatus

Class	**DiagnosticRequestEmissionRelatedDTCPermanentStatusClass**			
Note	This meta-class represents the ability to define common properties for all instances of the "Request Emission Related DTC Permanent Status" OBD diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRequestEmissionRelatedDTCPermanentStatuss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.148: DiagnosticRequestEmissionRelatedDTCPermanentStatusClass

4.3.10 UDS Diagnostic Services for supporting WWH-OBD

[TPS_DEXT_01133] Support for WWH-OBD within the diagnostic extract ⌈A support for WWH-OBD (see [20, ISO 27145-3]) within the diagnostic extract involves the usage of the following UDS services and their respective subfunctions:

DiagnosticReadDataByIdentifier (0x22) where the value of attribute DiagnosticDataIdentifier.id inside the interval defined by the OBD range, i.e.:

- F400-F4FF

- F600-F6FF
- F800-F8FF

Please note that the usage of this diagnostic service in an implementation of WWH-OBD corresponds to the existence of a DiagnosticValueNeeds on the side of an AtomicSwComponentType that interacts with the diagnostic service.

DiagnosticRoutineControl (0x31) where the value of attribute DiagnosticRoutine.id is inside the interval defined by the OBD range, i.e. E000-E0FF. Please note that the usage of this diagnostic service in an implementation of WWH-OBD corresponds to the existence of a DiagnosticRoutineNeeds on the side of an AtomicSwComponentType that implements routines executed within the diagnostic service.

DiagnosticClearDiagnosticInformation (0x14)

DiagnosticReadDTCInformation (0x19) with the subfunctions (0x4, 0x6, 0x42)

⌋

When used in a WWH-OBD environment, meta-class DiagnosticRoutine utilizes the attribute routineInfo to support the configuration of the diagnostic response. In a pure UDS environment (except in an application of [18, ISO 26021-2]) the attribute has no semantics.

4.4 Diagnostic Event Handling

4.4.1 Introduction

This sub-chapter describes the meta-model elements that define the handling of and the functionality around diagnostic events.

In a standard AUTOSAR Basic Software architecture for the *AUTOSAR classic platform*, the definitions based on the model elements described in this sub-chapter are realized by the Diagnostic Event Manager (Dem) module.

The following figure gives an overview on the model elements related to the diagnostic event functionality.

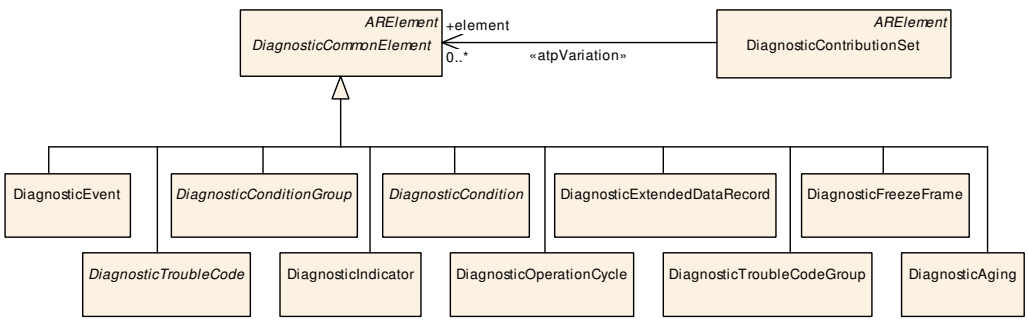

Figure 4.55: Common model elements relevant for the Dem

For the definition of the diagnostic event functionality, a number of model elements are derived from DiagnosticCommonElement. These elements are described in the following sub-chapters.

4.4.2 Diagnostic Event

4.4.2.1 Overview

[TPS_DEXT_01083] Semantics of a DiagnosticEvent ⌈A DiagnosticEvent - the atomic unit handled by the Dem module - has to be defined together with its properties which affect the event handling behavior and possible interfaces to software-components.⌋

Figure 4.56 depicts the definition of DiagnosticEvent together with its properties.

The DiagnosticExtract allows the definition of an arbitrary number of DiagnosticEvents.

Although the exchange of a DiagnosticExtract between companies usually involves DiagnosticEvents related to SWC functionality, the event kind BSW is also supported in order to enable definition of the handling of BSW events (e.g. definition of associated DiagnosticTroubleCode).

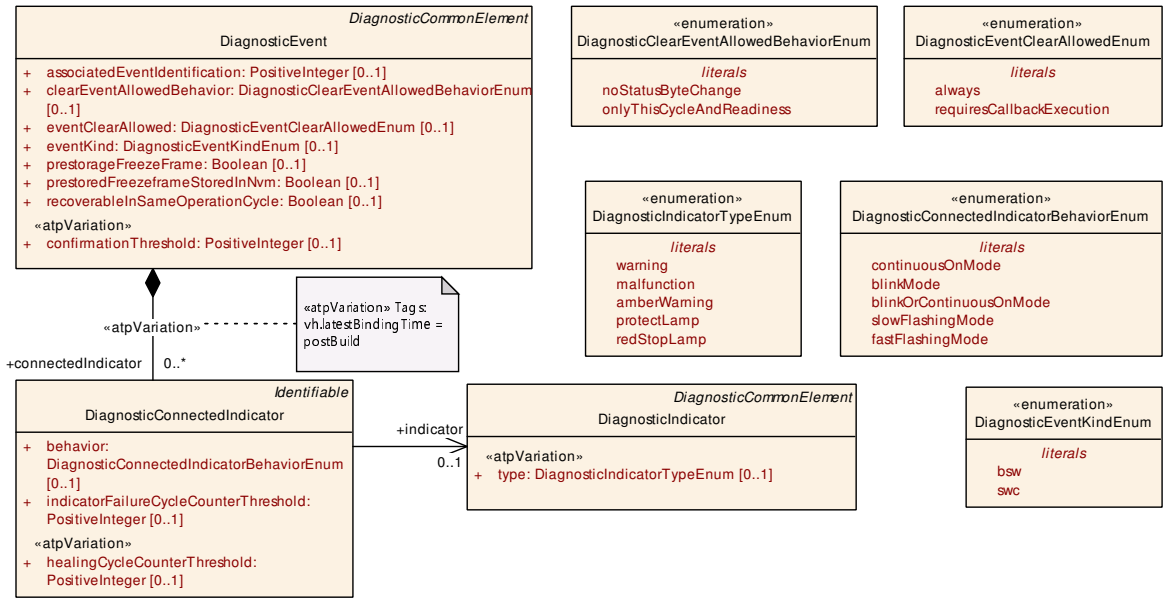

Figure 4.56: Modeling of DiagnosticEvent

[TPS_DEXT_03011] Clearing request for a DiagnosticEvent ⌈Furthermore, a clearing request for a DiagnosticEvent might require invocation of a callback to a SWC in order to allow or to prohibit the clearing operation.

The expectation for this callback interface can be expressed using the attribute eventClearAllowed:

- `always` indicates that a clearing request for the `DiagnosticEvent` **shall unconditionally be executed**.
- In the case of `requiresCallbackExecution`, the **execution of a callback shall decide whether the clearing is permitted**.

  In other words, the implementation of this decision is up to the developer of the corresponding `AtomicSwComponentType`.

  The latter shall define a `SwcServiceDependency` with appropriate `DiagnosticEventNeeds` and a `RoleBasedPortAssignment` where the value of the attribute `role` is set to `CallbackClearEventAllowed`.

⌋

Class	**DiagnosticEvent**			
Note	This element is used to configure DiagnosticEvents. **Tags:** atp.recommendedPackage=DiagnosticEvents			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
associated Event Identification	PositiveInteger	0..1	attr	This attribute represents the identification number that is associated with the enclosing DiagnosticEvent and allows to identify it when placed into a snapshot record or extended data record storage. This value can be reported as internal data element in snapshot records or extended data records.
clearEvent Allowed Behavior	DiagnosticClearEvent AllowedBehaviorEnum	0..1	attr	This attribute defines the resulting UDS status byte for the related event, which shall not be cleared according to the ClearEventAllowed callback
confirmation Threshold	PositiveInteger	0..1	attr	This attribute defines the number of operation cycles with a failed result before a confirmed DTC is set to 1. The semantic of this attribute is a by "1" increased value compared to the confirmation threshold of the "trip counter" mentioned in ISO 14229-1 in figure D.4. A value of "1" defines the immediate confirmation of the DTC along with the first reported failed. This is also sometimes called "zero trip DTC". A value of "2" defines a DTC confirmation in the operation cycle after the first occurred failed. A value of "2" is typically used in the US for OBD DTC confirmation. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
connected Indicator	DiagnosticConnected Indicator	*	aggr	Event specific description of Indicators. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=connectedIndicator, connected Indicator.shortName, connectedIndicator.variation Point.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	**DiagnosticEvent**			
eventClear Allowed	DiagnosticEventClear AllowedEnum	0..1	attr	This attribute defines whether the Dem has access to a "ClearEventAllowed" callback. This Attribute is only used by the AUTOSAR Classic Platform.
eventKind	DiagnosticEventKind Enum	0..1	attr	This attribute is used to distinguish between SWC and BSW events. This Attribute is only used by the AUTOSAR Classic Platform.
prestorage FreezeFrame	Boolean	0..1	attr	This attribute describes whether the Prestorage of Freeze Frames is supported by the assigned event or not. true: Prestorage of FreezeFrames is supported fFalse: Prestorage of FreezeFrames is not supported
prestored Freezeframe StoredInNvm	Boolean	0..1	attr	If the Event uses a prestored freeze-frame (using the operations PrestoreFreezeFrame and ClearPrestored FreezeFrame of the service interface DiagnosticMonitor) this attribute indicates if the Event requires the data to be stored in non-volatile memory. TRUE = Dem shall store the prestored data in non-volatile memory, FALSE = Data can be lost at shutdown (not stored in Nvm)
recoverableIn SameOperation Cycle	Boolean	0..1	attr	If the attribute is set to true then reporting PASSED will reset the indication of a failed test in the current operation cycle. If the attribute is set to false then reporting PASSED will be ignored and not lead to a reset of the indication of a failed test.

Table 4.149: DiagnosticEvent

Enumeration	**DiagnosticClearEventAllowedBehaviorEnum**
Note	This enumeration defines the possible behavior for clear event allowed
Aggregated by	DiagnosticEvent.clearEventAllowedBehavior
Literal	*Description*
noStatusByte Change	The event status byte keeps unchanged. **Tags:** atp.EnumerationLiteralIndex=0
onlyThisCycleAnd Readiness	The OperationCycle and readiness bits of the event status byte are reset. **Tags:** atp.EnumerationLiteralIndex=1

Table 4.150: DiagnosticClearEventAllowedBehaviorEnum

Class	**RoleBasedPortAssignment**			
Note	This class specifies an assignment of a role to a particular service port (RPortPrototype or PPort Prototype) of an AtomicSwComponentType. With this assignment, the role of the service port can be mapped to a specific ServiceNeeds element, so that a tool is able to create the correct connector.			
Base	*ARObject*			
Aggregated by	NvBlockDescriptor.clientServerPort, SwcServiceDependency.assignedPort			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
portPrototype	PortPrototype	0..1	ref	Service PortPrototype used in the assigned role. This PortPrototype shall either belong to the same AtomicSw ComponentType as the SwcInternalBehavior which owns the ServiceDependency or to the same NvBlockSw ComponentType as the NvBlockDescriptor. **Tags:** atp.Splitkey=portPrototype

▽

△

Class	**RoleBasedPortAssignment**			
role	Identifier	0..1	attr	This is the role of the assigned Port in the given context. The value shall be a shortName of the Blueprint of a Port Interface as standardized in the Software Specification of the related AUTOSAR Service.

Table 4.151: RoleBasedPortAssignment

[TPS_DEXT_01085] `DiagnosticEvent` can be connected to one or multiple indicators ⌈A `DiagnosticEvent` can be connected to one or multiple indicators (modeled by means of aggregating `DiagnosticIndicator` in the role `connectedIndicator`) of a certain type and with certain behavior.⌋

Class	**DiagnosticConnectedIndicator**			
Note	Description of indicators that are defined per DiagnosticEvent.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticEvent.connectedIndicator			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
behavior	DiagnosticConnectedIndicatorBehaviorEnum	0..1	attr	Behavior of the linked indicator.
healingCycleCounterThreshold	PositiveInteger	0..1	attr	This attribute defines the number of healing cycles for the WarningIndicatorOffCriteria **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
indicator	DiagnosticIndicator	0..1	ref	Reference to the used indicator. **Tags:** atp.Splitkey=indicator
indicatorFailureCycleCounterThreshold	PositiveInteger	0..1	attr	This attribute defines the number of failure cycles for the WarningIndicatorOnCriteria. Please note that this attribute is not relevant for the Adaptive Platform.

Table 4.152: DiagnosticConnectedIndicator

[constr_1829] Existence of reference `DiagnosticConnectedIndicator.indicator`

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈For each `DiagnosticConnectedIndicator`, the reference to `DiagnosticIndicator` in the role `indicator` shall exist.⌋

Enumeration	**DiagnosticEventClearAllowedEnum**
Note	Denotes whether clearing of events is allowed.
Aggregated by	DiagnosticEvent.eventClearAllowed
Literal	***Description***
always	The clearing is allowed unconditionally. **Tags:** atp.EnumerationLiteralIndex=0

▽

△

Enumeration	**DiagnosticEventClearAllowedEnum**
requiresCallback Execution	In case the clearing of a Diagnostic Event has to be allowed or prohibited through the SWC interface CallbackClearEventAllowed, the SWC has to indicate this by defining appropriate ServiceNeeds (i.e. DiagnosticEventNeeds). **Tags:** atp.EnumerationLiteralIndex=2

Table 4.153: DiagnosticEventClearAllowedEnum

Enumeration	**DiagnosticEventKindEnum**
Note	Applicability of the diagnostic event.
Aggregated by	DiagnosticEvent.eventKind
Literal	***Description***
bsw	The event is assigned to a BSW module. **Tags:** atp.EnumerationLiteralIndex=0
swc	The event is assigned to a SWC. **Tags:** atp.EnumerationLiteralIndex=1

Table 4.154: DiagnosticEventKindEnum

Enumeration	**DiagnosticConnectedIndicatorBehaviorEnum**
Note	Behavior of the indicator.
Aggregated by	DiagnosticConnectedIndicator.behavior
Literal	***Description***
blinkMode	The indicator blinks when the event has status FAILED. **Tags:** atp.EnumerationLiteralIndex=0
blinkOrContinuous OnMode	The indicator is active and blinks when the event has status FAILED. **Tags:** atp.EnumerationLiteralIndex=1
continuousOnMode	The indicator is active when the event has status FAILED. **Tags:** atp.EnumerationLiteralIndex=2
fastFlashingMode	Flash Indicator Lamp should be set to "Fast Flash". **Tags:** atp.EnumerationLiteralIndex=3
slowFlashingMode	Flash Indicator Lamp should be set to "Slow Flash". **Tags:** atp.EnumerationLiteralIndex=4

Table 4.155: DiagnosticConnectedIndicatorBehaviorEnum

4.4.2.2 Textual description

[TPS_DEXT_01067] Textually formulated content attached to `DiagnosticEvent`
⌈The definition of a `DiagnosticEvent` also consists of textually formulated content that is formalized in structure but cannot be formalized in content.

The purpose of this content is to define e.g. a mature condition that relates to the specific `DiagnosticEvent`.⌋

[TPS_DEXT_01068] Textual description with respect to the `DiagnosticEvent`
⌈A textual description that has the character of requirements with respect to the

DiagnosticEvent shall be provided by means of the meta-class StructuredReq, i.e. by means of introduction.structuredReq.⌋

Class	StructuredReq			
Note	This represents a structured requirement. This is intended for a case where specific requirements for features are collected. Note that this can be rendered as a labeled list.			
Base	*ARObject*, *DocumentViewSelectable*, *Identifiable*, *MultilanguageReferrable*, *Paginateable*, *Referrable*, *Traceable*			
Aggregated by	DocumentationBlock.structuredReq			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
appliesTo	StandardNameEnum	*	attr	This attribute represents the platform the requirement is assigned to. **Tags:** xml.namePlural=APPLIES-TO-DEPENDENCIES xml.sequenceOffset=25
conflicts	DocumentationBlock	0..1	aggr	This represents an informal specification of conflicts. **Tags:** atp.Splitkey=conflicts xml.sequenceOffset=40
date	DateTime	0..1	attr	This represents the date when the requirement was initiated. **Tags:** xml.sequenceOffset=5
dependencies	DocumentationBlock	0..1	aggr	This represents an informal specification of dependencies. Note that upstream tracing should be formalized in the property trace provided by the superclass Traceable. **Tags:** atp.Splitkey=dependencies xml.sequenceOffset=30
description	DocumentationBlock	0..1	aggr	This represents the general description of the requirement. **Tags:** atp.Splitkey=description xml.sequenceOffset=10
importance	String	0..1	attr	This allows to represent the importance of the requirement. **Tags:** xml.sequenceOffset=8
issuedBy	String	0..1	attr	This represents the person, organization or authority which issued the requirement. **Tags:** xml.sequenceOffset=6
rationale	DocumentationBlock	0..1	aggr	This represents the rationale of the requirement. **Tags:** atp.Splitkey=rationale xml.sequenceOffset=20
remark	DocumentationBlock	0..1	aggr	This represents an informal remark. Note that this is not modeled as annotation, since these remark is still essential part of the requirement. **Tags:** atp.Splitkey=remark xml.sequenceOffset=60

▽

△

Class	StructuredReq			
supporting Material	DocumentationBlock	0..1	aggr	This represents an informal specification of the supporting material. **Tags:** atp.Splitkey=supportingMaterial xml.sequenceOffset=50
testedItem	Traceable	*	ref	This association represents the ability to trace on the same specification level. This supports for example the of acceptance tests. **Tags:** atp.Splitkey=testedItem xml.sequenceOffset=70
type	String	0..1	attr	This attribute allows to denote the type of requirement to denote for example is it an "enhancement", "new feature" etc. **Tags:** xml.sequenceOffset=7
useCase	DocumentationBlock	0..1	aggr	This describes the relevant use cases. Note that formal references to use cases should be done in the trace relation. **Tags:** atp.Splitkey=useCase xml.sequenceOffset=35

Table 4.156: StructuredReq

Enumeration	StandardNameEnum
Note	This enumeration lists all allowed standard abbreviations.
Aggregated by	AppliedStandard.appliesTo, StructuredReq.appliesTo
Literal	*Description*
AP	This values represents the Adaptive Platform. **Tags:** atp.EnumerationLiteralIndex=0
CP	This Value represents the Classic Platform. **Tags:** atp.EnumerationLiteralIndex=1
FO	This values represents the Foundation. **Tags:** atp.EnumerationLiteralIndex=2

Table 4.157: StandardNameEnum

For more details regarding the modeling of the semi-formal text please refer to Figure 4.4.

[TPS_DEXT_01069] Standardized values of `DiagnosticEvent.introduction.structuredReq` ⌈The following possible values of `DiagnosticEvent.introduction.structuredReq` are standardized by AUTOSAR:

- `DIAG_EVENT_MON_COND`: this value describes the monitoring condition of the corresponding `DiagnosticEvent`.
- `DIAG_EVENT_MON_TYPE`: this value describes the monitoring type of the corresponding `DiagnosticEvent`.
- `DIAG_EVENT_MON_RATE`: this value describes the monitoring rate for the corresponding `DiagnosticEvent`.

- DIAG_EVENT_MAT_COND: this value describes a mature condition of the DiagnosticEvent.
- DIAG_EVENT_DEMAT_COND: this value describes a de-mature condition of the DiagnosticEvent.
- DIAG_EVENT_AGING: this value describes the behavior of the DiagnosticEvent regarding aging.
- DIAG_EVENT_LIMP_IN_ACT: this value describes the associated limp-in action for the DiagnosticEvent.
- DIAG_EVENT_MAT_TIME: this value describes the mature time for the corresponding DiagnosticEvent, i.e. how long or how often the fault shall exist.
- DIAG_EVENT_DEMAT_TIME: this value describes the de-mature time for the corresponding DiagnosticEvent, i.e. how often shall the OK conditions be fulfilled.

⌋

Please find an example definition of a mature condition in Chapter B.3.

4.4.2.3 Associated Event Identification

[TPS_DEXT_01151] Semantics of attribute DiagnosticEvent.associatedEventIdentification ⌈Use cases exist where DiagnosticEvents are not associated with the DiagnosticTroubleCode but the occurrence of the event shall nonetheless be reported as part of the troubleCodeProps.snapshotRecordContent.diagnosticDataIdentifier of a given DiagnosticTroubleCodeUds.

For this purpose, it is necessary that the tester can identify the occurrence of the DiagnosticEvent by means of the value of the attribute DiagnosticEvent.associatedEventIdentification.⌋

[constr_1768] Existence of attribute DiagnosticEvent.associatedEventIdentification

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈Attribute DiagnosticEvent.associatedEventIdentification shall exist if

- the respective DiagnosticEvent is mapped to a DiagnosticTroubleCodeUds and
- the reference in the role DiagnosticTroubleCodeUds.troubleCodeProps.extendedDataRecord exists and
- the referenced DiagnosticExtendedDataRecord aggregates at least one element that is also referenced by a DiagnosticEdrDataProviderMapping

that has attribute dataProvider set to the value DEM_EVENT_ASSOCIATED_IDENTIFICATION.

⌋

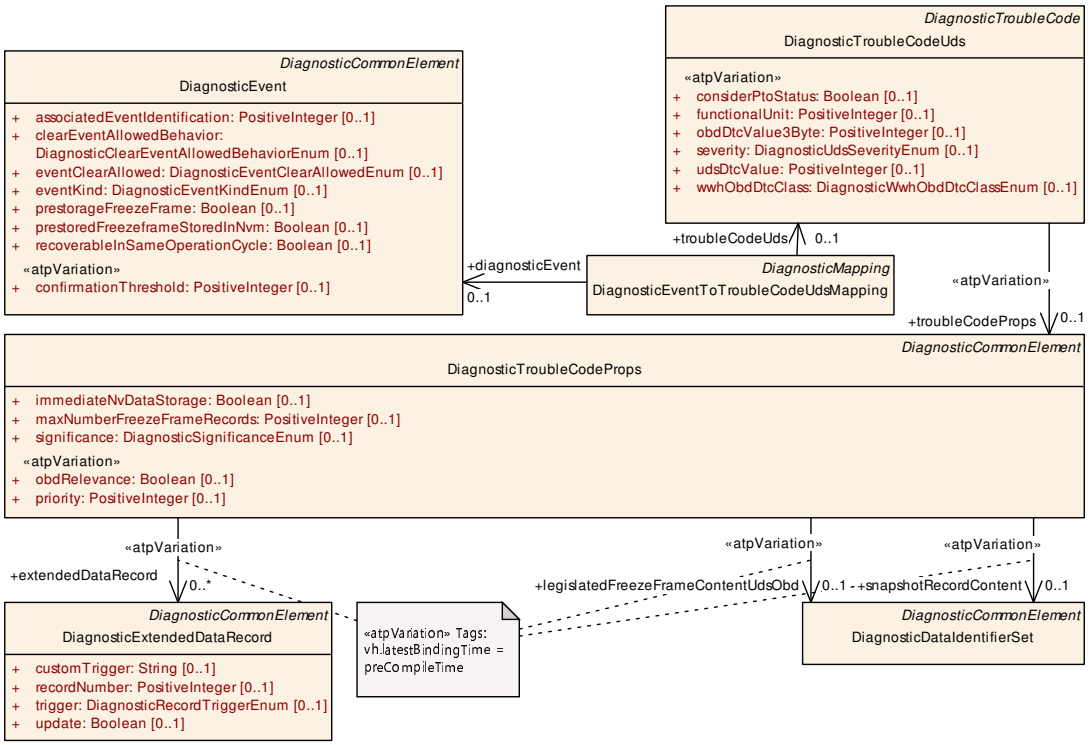

Figure 4.57: Modeling of associated event identification

The use case described in [TPS_DEXT_01151] and [constr_1768] is depicted in Figure 4.57.

Please note that [constr_1768] only describes the minimum viable scenario for the usage of the attribute DiagnosticEvent.associatedEventIdentification.

It is also possible that more than one DiagnosticExtendedDataRecordElement exists that is referenced by a DiagnosticEdrDataProviderMapping where the attribute dataProvider is set to the value DEM_EVENT_ASSOCIATED_IDENTIFICATION.

4.4.3 Diagnostic Trouble Code

DiagnosticTroubleCodes (i.e. the ECU external view on diagnostic events) are defined together with their properties and mapped to DiagnosticEvents using DiagnosticEventToTroubleCodeUdsMapping.

[TPS_DEXT_03012] Three kinds of DTCs ⌈There are three kinds of DTCs represented as specializations of DiagnosticTroubleCode:

- the UDS and OBD-on-UDS DTCs (DiagnosticTroubleCodeUds)

- the OBD-II-related DTCs (DiagnosticTroubleCodeObd)
- the [21, SAE J1939-73]-related DTCs (DiagnosticTroubleCodeJ1939)

Properties individual to such a DTC specialization are modeled as attributes of DiagnosticTroubleCodeUds, DiagnosticTroubleCodeObd and DiagnosticTroubleCodeJ1939, respectively.⌋

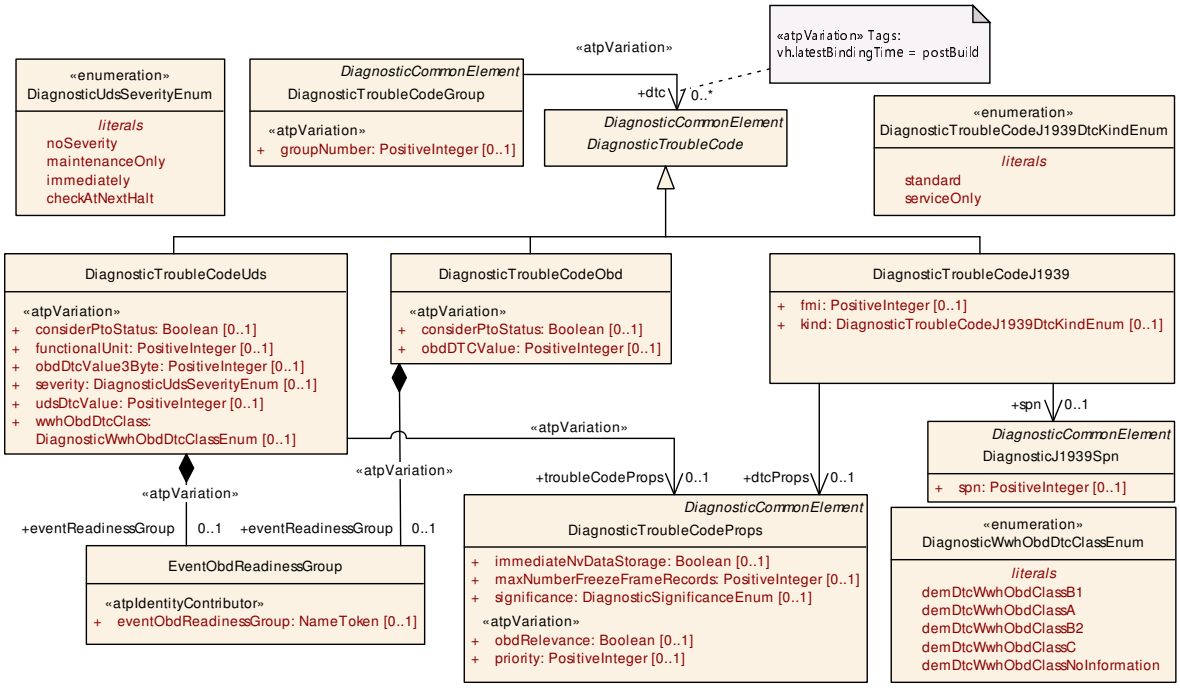

Figure 4.58: Modeling of DiagnosticTroubleCode

If it is intended to implement OBD-on-UDS, then the definition of a DiagnosticTroubleCodeUds is sufficient. On the other hand, if the intention is to utilize OBD-II, then a DiagnosticTroubleCodeObd needs to be defined.

Class	**DiagnosticTroubleCodeUds**			
Note	This element is used to describe diagnostic trouble codes (DTCs). **Tags:** atp.recommendedPackage=DiagnosticTroubleCodes			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticTroubleCode*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
considerPto Status	Boolean	0..1	attr	This attribute describes the affection of the event by the Dem PTO handling. true: the event is affected by the Dem PTO handling. false: the event is not affected by the Dem PTO handling. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

△

Class	DiagnosticTroubleCodeUds			
eventReadiness Group	EventObdReadiness Group	0..1	aggr	This attribute specifies the Event OBD Readiness group for PID $01 and PID $41 computation. This attribute is only applicable for emission-related ECUs. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=eventReadinessGroup, eventReadiness Group.eventObdReadinessGroup, eventReadiness Group.variationPoint.shortLabel vh.latestBindingTime=postBuild
functionalUnit	PositiveInteger	0..1	attr	This attribute specifies a 1-byte value which identifies the corresponding basic vehicle / system function which reports the DTC. This parameter is necessary for the report of severity information. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
obdDtcValue3 Byte	PositiveInteger	0..1	attr	3 Byte OBD DTC value based on the definition from SAE J2012. The existence of this attribute is only required if separated UDS and OBD DTC values are used for SAE J1979-2. If this attribute does not exist, then UDS DTC values are used with J1979-2. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
severity	DiagnosticUdsSeverity Enum	0..1	attr	DTC severity according to ISO 14229-1. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
troubleCode Props	DiagnosticTroubleCode Props	0..1	ref	Defined properties associated with the DemDTC. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=troubleCodeProps, troubleCode Props.diagnosticTroubleCodeProps, troubleCode Props.variationPoint.shortLabel vh.latestBindingTime=postBuild
udsDtcValue	PositiveInteger	0..1	attr	Unique Diagnostic Trouble Code value for UDS. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
wwhObdDtc Class	DiagnosticWwhObdDtc ClassEnum	0..1	attr	This attribute is used to identify (if applicable) the corresponding severity class of an WWH-OBD DTC. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table 4.158: DiagnosticTroubleCodeUds

Class	DiagnosticTroubleCodeObd			
Note	This element is used to define OBD-relevant DTCs. **Tags:** atp.recommendedPackage=DiagnosticTroubleCodes			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticTroubleCode*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	**DiagnosticTroubleCodeObd**			
considerPto Status	Boolean	0..1	attr	This attribute describes the affection of the event by the Dem PTO handling. true: the event is affected by the Dem PTO handling. false: the event is not affected by the Dem PTO handling. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
eventReadiness Group	EventObdReadiness Group	0..1	aggr	This aggregation allows for the variant definition of the attribute eventObdReadinessGroup. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=eventReadinessGroup, eventReadiness Group.eventObdReadinessGroup, eventReadiness Group.variationPoint.shortLabel vh.latestBindingTime=postBuild
obdDTCValue	PositiveInteger	0..1	attr	Unique Diagnostic Trouble Code value for OBD. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table 4.159: DiagnosticTroubleCodeObd

But the latter is not able to function on its own. It always needs to be associated with a DiagnosticTroubleCodeUds by means of a DiagnosticTroubleCodeUdsToTroubleCodeObdMapping.

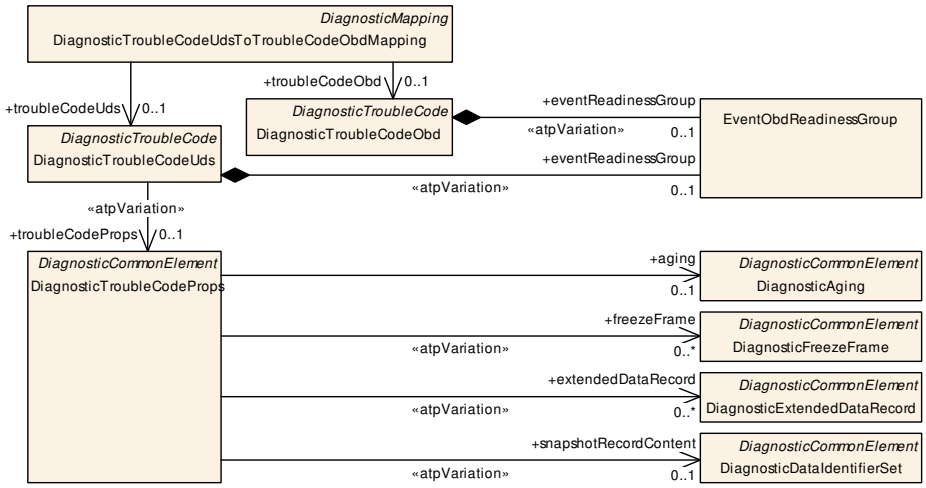

Figure 4.59: Mapping of OBD DTC to UDS DTC

[TPS_DEXT_01201] Semantics of meta-class DiagnosticTroubleCodeUdsToTroubleCodeObdMapping ⌈The meta-class DiagnosticTroubleCodeUdsToTroubleCodeObdMapping is used to associate a DiagnosticTroubleCodeObd with a DiagnosticTroubleCodeUds.

The reason for this dependency is that the DiagnosticTroubleCodeUds provides (via the associated DiagnosticTroubleCodeProps) features like extended data records and freeze frames that are not available for DiagnosticTroubleCodeObd.⌋

[constr_10750] Existence of DiagnosticTroubleCodeUdsToTroubleCodeObdMapping

Imposition time: IT_DiagDes

⌈For each DiagnosticTroubleCodeObd, a DiagnosticTroubleCodeUdsToTroubleCodeObdMapping shall exist that references the DiagnosticTroubleCodeObd in the role troubleCodeObd.⌋

As depicted in Figure 4.59, both DiagnosticTroubleCodeUds and DiagnosticTroubleCodeObd are able to reference EventObdReadinessGroup. However, if a DiagnosticTroubleCodeUdsToTroubleCodeObdMapping exists, then only one reference to EventObdReadinessGroup is required.

[constr_10751] Existence of reference to EventObdReadinessGroup

Imposition time: IT_DiagDes

⌈For each DiagnosticTroubleCodeUdsToTroubleCodeObdMapping, **only** the referenced troubleCodeObd shall aggregate a EventObdReadinessGroup in the role eventReadinessGroup while the troubleCodeUds shall not aggregate a EventObdReadinessGroup in the role eventReadinessGroup.⌋

Class	**DiagnosticTroubleCodeUdsToTroubleCodeObdMapping**			
Note	This meta-class represents the ability to associate an OBD trouble code to a UDS trouble code. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
troubleCodeObd	DiagnosticTroubleCodeObd	0..1	ref	This represents the OBD DTC referenced in the mapping between UDS and OBD DTCs. **Tags:** atp.Splitkey=troubleCodeObd
troubleCodeUds	DiagnosticTroubleCodeUds	0..1	ref	This represents the UDS DTC referenced in the mapping between UDS and OBD DTCs. **Tags:** atp.Splitkey=troubleCodeUds

Table 4.160: DiagnosticTroubleCodeUdsToTroubleCodeObdMapping

[constr_10523] Existence of role DiagnosticTroubleCodeUdsToTroubleCodeObdMapping.troubleCodeUds

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticTroubleCodeUdsToTroubleCodeObdMapping, the reference in the role troubleCodeUds shall exist.⌋

[constr_10524] Existence of role DiagnosticTroubleCodeUdsToTroubleCodeObdMapping.troubleCodeObd

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticTroubleCodeUdsToTroubleCodeObdMapping, the reference in the role troubleCodeObd shall exist.⌋

[constr_1757] Existence of the attribute DiagnosticTroubleCodeUds.udsDtcValue

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticTroubleCodeUds, attribute udsDtcValue shall exist.⌋

[constr_10088] Relation between event and DTC without event combination

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈If the primitive attribute DiagnosticCommonProps.typeOfEventCombinationSupported is not configured, then all DiagnosticTroubleCodeUds that refer to a DiagnosticTroubleCodeProps in the role troubleCodeProps that in turn refers to a DiagnosticMemoryDestination in the role diagnosticMemory shall only be referenced by at most one DiagnosticEventToTroubleCodeUdsMapping.⌋

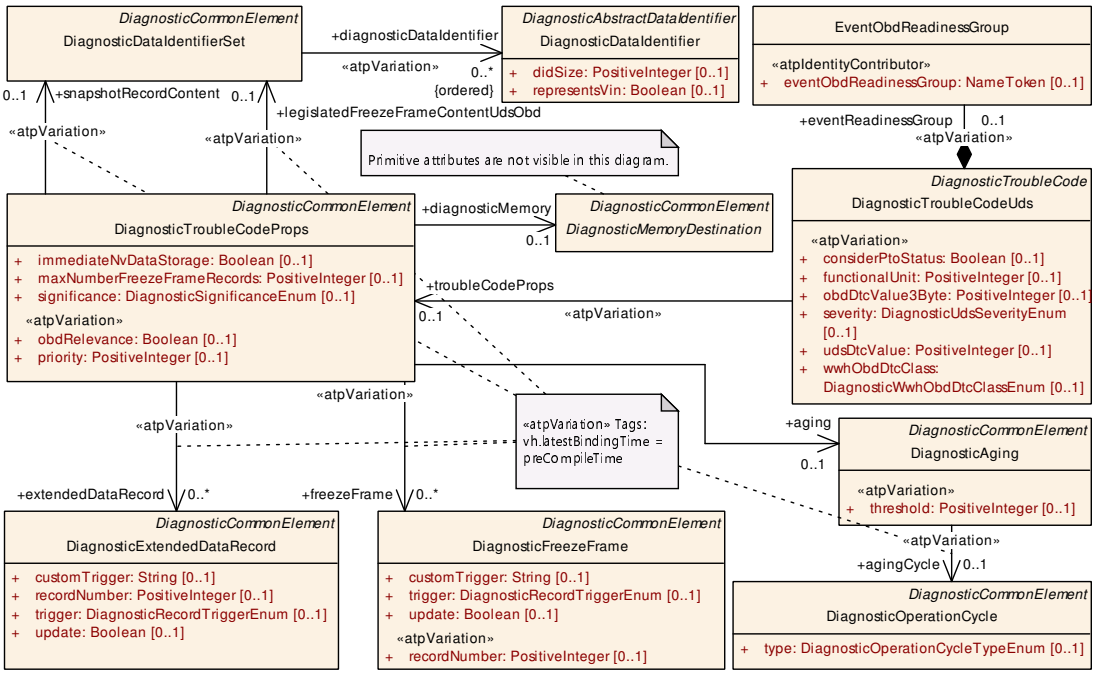

Figure 4.60: Modeling of DiagnosticTroubleCodeUds

Class	**EventObdReadinessGroup**			
Note	This meta-class represents the ability to define the value of attribute eventObdReadinessGroup. It is only introduced to allow for a variant modeling of this attribute.			
Base	*ARObject*			
Aggregated by	DiagnosticTroubleCodeObd.eventReadinessGroup, DiagnosticTroubleCodeUds.eventReadinessGroup			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
eventObd Readiness Group	NameToken	0..1	attr	This attribute specifies the Event OBD Readiness group for PID $01 and PID $41 computation. This attribute is only applicable for emission-related ECUs. **Stereotypes:** atpIdentityContributor

Table 4.161: EventObdReadinessGroup

[constr_1758] Existence of primitive attribute `DiagnosticTroubleCodeObd.obdDTCValue`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticTroubleCodeObd`, attribute `obdDTCValue` shall exist.⌋

[TPS_DEXT_03013] Common properties of a DTC ⌈Properties that are often common for a group of `DiagnosticTroubleCodeUds` elements are modeled as attributes of `DiagnosticTroubleCodeProps`.⌋

[constr_1349] Value of `udsDtcValue` shall be unique

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈The value of `DiagnosticTroubleCodeUds.udsDtcValue` shall be unique for all `DiagnosticTroubleCodeUds` that refer to the same `DiagnosticMemoryDestination` via the reference `DiagnosticTroubleCodeUds.troubleCodeProps.diagnosticMemory`.⌋

Class	***DiagnosticTroubleCode*** (abstract)			
Note	A diagnostic trouble code defines a unique identifier that is shown to the diagnostic tester.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticTroubleCodeJ1939, DiagnosticTroubleCodeObd, DiagnosticTroubleCodeUds			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.162: DiagnosticTroubleCode

[TPS_DEXT_03014] Semantics of `DiagnosticTroubleCodeGroup` ⌈The `DiagnosticTroubleCodeGroup` element is used to define groups of `DTC`s that belong together. Each `DiagnosticTroubleCodeGroup` has its own `groupNumber` value assigned.⌋

Class	**DiagnosticTroubleCodeGroup**			
Note	The diagnostic trouble code group defines the DTCs belonging together and thereby forming a group. **Tags:** atp.recommendedPackage=DiagnosticTroubleCodes			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dtc	DiagnosticTroubleCode	*	ref	This represents the collection of DiagnosticTroubleCodes defined by this DiagnosticTroubleCodeGroup. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dtc, dtc.diagnosticTroubleCode, dtc.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	**DiagnosticTroubleCodeGroup**			
groupNumber	PositiveInteger	0..1	attr	This represents the base number of the DTC group. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table 4.163: DiagnosticTroubleCodeGroup

[constr_1830] Existence of `DiagnosticTroubleCodeGroup.groupNumber`

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈For each `DiagnosticTroubleCodeGroup`, attribute `groupNumber` shall exist.⌋

[constr_1350] Value of `DiagnosticTroubleCodeGroup.groupNumber` shall be unique

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈The value of `DiagnosticTroubleCodeGroup.groupNumber` shall be unique to any other `DTC` and `DTC` group value.⌋

[constr_1351] Value of `DiagnosticTroubleCodeGroup.groupNumber`

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈To be compliant to ISO, the value of `DiagnosticTroubleCodeGroup.groupNumber` shall be set as defined in [17, ISO 14229-1].⌋

[TPS_DEXT_03000] ISO 14229-1 reserves values of `DiagnosticTroubleCodeGroup.groupNumber` ⌈Any values other than those mentioned in [constr_1351] are reserved by [17, ISO 14229-1].⌋

[constr_1352] Existence of `maxNumberFreezeFrameRecords` vs. `freezeFrame`

Imposition time: CP: `IT_DiagDes`, AP: `IT_DiagDes`

⌈If the attribute `DiagnosticTroubleCodeProps.maxNumberFreezeFrameRecords` exists, then the attribute `DiagnosticTroubleCodeProps.freezeFrame` shall not exist or vice versa.⌋

[constr_1353] Applicability of [constr_1352]

Imposition time: CP: `IT_DiagDes`, AP: `IT_DiagDes`

⌈[constr_1352] shall apply in the identical way (either one or the other attribute shall exist) for all `DiagnosticTroubleCodeProps` within the context of all `DiagnosticContributionSet`s of `category` DIAGNOSTIC_ECU_EXTRACT that refer to the same `EcuInstance`.⌋

[constr_1354] Existence of the reference in the role `DiagnosticTroubleCodeProps.snapshotRecordContent`

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈If one of the attributes `DiagnosticTroubleCodeProps.maxNumberFreezeFrameRecords` or `DiagnosticTroubleCodeProps.freezeFrame` exists then the

reference in the role DiagnosticTroubleCodeProps.snapshotRecordContent shall exist.⌋

[TPS_DEXT_01064] Textually formulated content attached to DiagnosticTroubleCode ⌈The definition of a DiagnosticTroubleCode also consists of textually formulated content that is formalized in structure but cannot be formalized in content.

The purpose of this content is to define e.g. an error text or the possible cause that relates to the specific DiagnosticTroubleCode.⌋

[TPS_DEXT_01065] Different approaches to provide semi-formal textual content attached to a DiagnosticTroubleCode ⌈There are different approaches to provide semi-formal textual content attached to a DiagnosticTroubleCode:

- A textual description that has the character of descriptions of the DiagnosticTroubleCode shall be provided by means of the meta-class TraceableText, i.e. by means of introduction.trace.
- A textual description that characterizes the DiagnosticTroubleCode with respect to the *ODX long name* shall be provided by means of the attribute longName.

⌋

Class	**TraceableText**			
Note	Represents a paragraph level text which can be referenced in order to establish tracing. It supports specific tracing of document items as specified in [TPS_STDT_00098]. The following approach applies: • shortName: represents the tag for tracing • longName: represents the headline • category: represents the kind of the tagged text			
Base	*ARObject*, *DocumentViewSelectable*, *Identifiable*, *MultilanguageReferrable*, *Paginateable*, *Referrable*, *Traceable*			
Aggregated by	DocumentationBlock.trace			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
text	DocumentationBlock	0..1	aggr	This represents the text to which the tag applies. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=30 xml.typeElement=false xml.typeWrapperElement=false

Table 4.164: TraceableText

Class	***MultilanguageReferrable*** (abstract)
Note	Instances of this class can be referred to by their identifier (while adhering to namespace borders). They also may have a longName. But they are not considered to contribute substantially to the overall structure of an AUTOSAR description. In particular it does not contain other Referrables.
Base	*ARObject*, *Referrable*

▽

△

Class	***MultilanguageReferrable*** (abstract)			
Subclasses	Caption, DefItem, DocumentationContext, *Identifiable*, SdgCaption, *TraceReferrable*, *Traceable*			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
longName	MultilanguageLongName	0..1	aggr	This specifies the long name of the object. Long name is targeted to human readers and acts like a headline. **Tags:** atp.Splitkey=longName

Table 4.165: MultilanguageReferrable

Class	**MultilanguageLongName**			
Note	This meta-class represents the ability to specify a long name which acts in the role of a headline. It is intended for human readers. Per language it should be around max 80 characters.			
Base	*ARObject*			
Aggregated by	AliasNameAssignment.label, *GeneralAnnotation*.label, *MultilanguageReferrable*.longName, Note.label, ValueGroup.label			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
l4	LLongName	*	aggr	This is the long name in one particular language. **Tags:** atp.Splitkey=l4 xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table 4.166: MultilanguageLongName

Class	«atpMixedString» **LLongName**			
Note	MixedContentForLongNames in one particular language. The language is denoted in the attribute l.			
Base	*ARObject*, *LanguageSpecific*, *MixedContentForLongName*			
Aggregated by	MultilanguageLongName.l4			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
blueprintValue	String	0..1	attr	This represents a description that documents how the value shall be defined when deriving objects from the blueprint. **Tags:** atp.Status=draft xml.attribute=true

Table 4.167: LLongName

For more details regarding the modeling of the semi-formal text please refer to Figure 4.4.

The usage of `TraceableText` and `StructuredReq` alone would not qualify as a semi-formal textual attachment. It is necessary to standardize the value of the `category` in order to get some level of semi-formal textual description.

[TPS_DEXT_01066] Standardized values of `DiagnosticTroubleCode.introduction.trace` ⌈The following possible values of `DiagnosticTroubleCode.introduction.trace` are standardized by AUTOSAR:

- **DIAG_DTC_ERROR_TEXT**: this value shall be used to describe an error text.

- **DIAG_DTC_REP_ACT**: this value describes the associated repair for the corresponding DiagnosticTroubleCode.
- **DIAG_DTC_CUS_PER_SYMP**: this value describes the possible customer perception symptom for the corresponding DiagnosticTroubleCode.
- **DIAG_DTC_POSS_CAUSE**: This value describes the possible cause for the corresponding DiagnosticTroubleCode.

⌋

An example of the definition of a semi-formal textual description by means of TraceableText can be found in Chapter B.4.

[constr_10522] OBD trouble code shall only be placed in primary fault memory

Imposition time: CP: IT_DiagDes

⌈If a DiagnosticTroubleCodeUds that is referenced in the role DiagnosticTroubleCodeUdsToTroubleCodeObdMapping.troubleCodeUds in turn references a DiagnosticTroubleCodeProps in the role troubleCodeProps, and if the DiagnosticTroubleCodeProps refers to a DiagnosticMemoryDestination in the role diagnosticMemory, then the referenced diagnosticMemory shall be a DiagnosticMemoryDestinationPrimary.⌋

Class	**DiagnosticTroubleCodeProps**			
Note	This element defines common Dtc properties that can be reused by different DTCs. **Tags:** atp.recommendedPackage=DiagnosticTroubleCodePropss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
aging	DiagnosticAging	0..1	ref	Reference to an aging algorithm in case that an aging/unlearning of the event is allowed. **Tags:** atp.Splitkey=aging
diagnostic Memory	DiagnosticMemory Destination	0..1	ref	Reference to the applicable DiagnosticMemory Destination. **Tags:** atp.Splitkey=diagnosticMemory
extendedData Record	DiagnosticExtended DataRecord	*	ref	Defines the links to an extended data class sampler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=extendedDataRecord, extendedData Record.diagnosticExtendedDataRecord, extendedData Record.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
freezeFrame	DiagnosticFreezeFrame	*	ref	Define the links to a freeze frame class sampler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=freezeFrame, freezeFrame.diagnosticFreeze Frame, freezeFrame.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**DiagnosticTroubleCodeProps**			
immediateNv DataStorage	Boolean	0..1	attr	Change description for Class immediateNvDataStorage in table "Table A.111: DiagnosticTroubleCodeProps": Switch to enable immediate storage triggering of an according event memory entry persistently to NVRAM. true: immediate non-volatile storage triggering on first occurrence and shutdown. false: immediate non-volatile storage triggering on shutdown.
legislated FreezeFrame ContentUdsObd	DiagnosticDataIdentifier Set	0..1	ref	This reference identifies the layout of legislated freeze frames used for emission related diagnostics over the UDS protocol such as OBDonUDS or WWH-OBD. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=legislatedFreezeFrameContentUdsObd, legislatedFreezeFrameContentUdsObd.diagnosticData IdentifierSet, legislatedFreezeFrameContentUds Obd.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
maxNumber FreezeFrame Records	PositiveInteger	0..1	attr	This attribute defines the number of according freeze frame records, which can maximal be stored for this event. Therefore all these freeze frame records have the same freeze frame class.
obdRelevance	Boolean	0..1	attr	The attribute controls whether the enclosing DTC is relevant for OBD. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
priority	PositiveInteger	0..1	attr	Priority of the event, in view of full event buffer. A lower value means higher priority. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
significance	DiagnosticSignificance Enum	0..1	attr	Significance of the event, which indicates additional information concerning fault classification and resolution.
snapshot RecordContent	DiagnosticDataIdentifier Set	0..1	ref	This represents the freeze frame layout as a set of DIDs. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=snapshotRecordContent, snapshotRecord Content.diagnosticDataIdentifierSet, snapshotRecord Content.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table 4.168: DiagnosticTroubleCodeProps

[constr_1831] Existence of `DiagnosticTroubleCodeProps.priority`

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈For each `DiagnosticTroubleCodeProps`, attribute `priority` shall exist.⌋

Enumeration	**DiagnosticSignificanceEnum**
Note	Significance level of a diagnostic event.
Aggregated by	DiagnosticTroubleCodeProps.significance
Literal	*Description*

▽

△

Enumeration	**DiagnosticSignificanceEnum**
fault	Failure, which affects the component/ECU itself. **Tags:** atp.EnumerationLiteralIndex=0
occurence	Issue, which indicates additional information concerning insufficient system behavior. **Tags:** atp.EnumerationLiteralIndex=1

Table 4.169: DiagnosticSignificanceEnum

Enumeration	**DiagnosticUdsSeverityEnum**
Note	Severity types for a DTC according to ISO 14229-1.
Aggregated by	DiagnosticTroubleCodeUds.severity
Literal	*Description*
checkAtNextHalt	Check at next halt. **Tags:** atp.EnumerationLiteralIndex=0
immediately	Check immediately. **Tags:** atp.EnumerationLiteralIndex=1
maintenanceOnly	Maintenance required. **Tags:** atp.EnumerationLiteralIndex=2
noSeverity	No severity information available. **Tags:** atp.EnumerationLiteralIndex=3

Table 4.170: DiagnosticUdsSeverityEnum

Class	**DiagnosticDataIdentifierSet**			
Note	This represents the ability to define a list of DiagnosticDataIdentifiers that can be reused in different contexts. **Tags:** atp.recommendedPackage=DiagnosticDataIdentifierSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
diagnosticDataIdentifier (ordered)	DiagnosticDataIdentifier	*	ref	Reference to an ordered list of Data Identifiers. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table 4.171: DiagnosticDataIdentifierSet

Enumeration	**DiagnosticWwhObdDtcClassEnum**
Note	This meta-class represents the ability to model severity classes of an WWH-OBD DTC.
Aggregated by	DiagnosticTroubleCodeUds.wwhObdDtcClass
Literal	*Description*
demDtcWwhObdClassA	This attribute represents the severity class A. **Tags:** atp.EnumerationLiteralIndex=0
demDtcWwhObdClassB1	This attribute represents the severity class B1. **Tags:** atp.EnumerationLiteralIndex=1

▽

△

Enumeration	**DiagnosticWwhObdDtcClassEnum**
demDtcWwhObd ClassB2	This attribute represents the severity class B2. **Tags:** atp.EnumerationLiteralIndex=2
demDtcWwhObd ClassC	This attribute represents the severity class C. **Tags:** atp.EnumerationLiteralIndex=3
demDtcWwhObd ClassNoInformation	This attribute represents the option to intentionally not describe a dedicated severity class of an WWH-OBD DTC. **Tags:** atp.EnumerationLiteralIndex=4

Table 4.172: DiagnosticWwhObdDtcClassEnum

4.4.4 Diagnostic Memory Destination

The "memory"-part of the fault memory is represented by the meta-class `DiagnosticMemoryDestination`. This meta-class is used to define the properties of a memory destination, e.g. the maximum number of entries, the displacement strategy, etc.

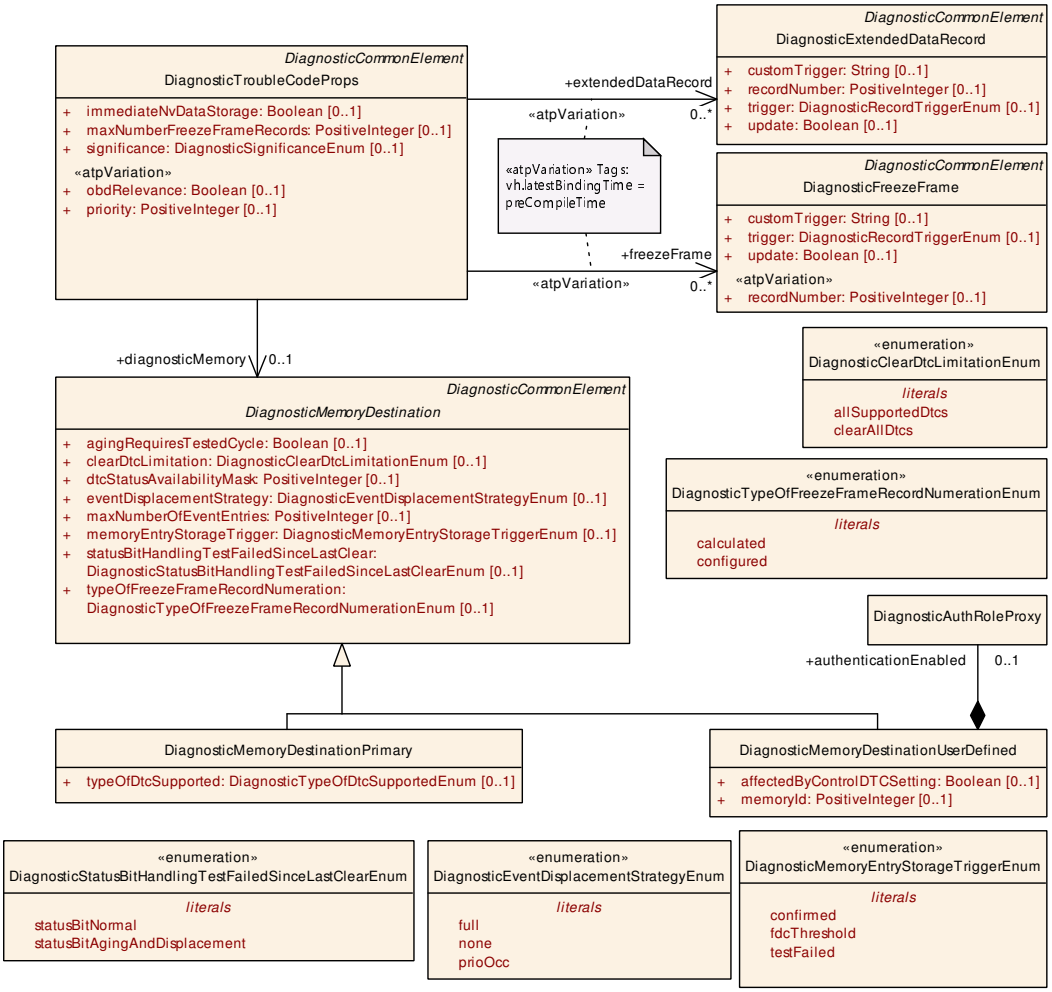

Figure 4.61: Modeling of `DiagnosticMemoryDestination`

[constr_1378] Value of DiagnosticMemoryDestinationUserDefined.memoryId

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈Within the scope of one DiagnosticContributionSet, no two (or more) DiagnosticMemoryDestinationUserDefineds shall exist that share the same value for attribute DiagnosticMemoryDestinationUserDefined.memoryId⌋

In other words, the value of the attribute DiagnosticMemoryDestinationUserDefined.memoryId shall be unique within any given DiagnosticExtract.

Class	***DiagnosticMemoryDestination*** (abstract)			
Note	This abstract meta-class represents a possible memory destination for a diagnostic event.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticMemoryDestinationPrimary, DiagnosticMemoryDestinationUserDefined			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
agingRequiresTestedCycle	Boolean	0..1	attr	Defines whether the aging cycle counter is processed every aging cycles or else only tested aging cycle are considered. If the attribute is set to TRUE: only tested aging cycle are considered for aging cycle counter. If the attribute is set to FALSE: aging cycle counter is processed every aging cycle. On the classic platform, the value of this attribute has to be identical for each DiagnosticMemoryDestination.
clearDtcLimitation	DiagnosticClearDtcLimitationEnum	0..1	attr	Defines the scope of the DEM_ClearDTC Api. On the classic platform, the value of this attribute has to be identical for each DiagnosticMemoryDestination.
dtcStatusAvailabilityMask	PositiveInteger	0..1	attr	Mask for the supported DTC status bits by the Dem.
eventDisplacementStrategy	DiagnosticEventDisplacementStrategyEnum	0..1	attr	This attribute defines, whether support for event displacement is enabled or not, and which displacement strategy is followed.
maxNumberOfEventEntries	PositiveInteger	0..1	attr	This attribute fixes the maximum number of event entries in the fault memory.
memoryEntryStorageTrigger	DiagnosticMemoryEntryStorageTriggerEnum	0..1	attr	Describes the trigger to allocate an event memory entry.
statusBitHandlingTestFailedSinceLastClear	DiagnosticStatusBitHandlingTestFailedSinceLastClearEnum	0..1	attr	This attribute defines, whether the aging and displacement mechanism shall be applied to the "TestFailedSinceLastClear" status bits. On the classic platform, the value of this attribute has to be identical for each DiagnosticMemoryDestination.
typeOfFreezeFrameRecordNumeration	DiagnosticTypeOfFreezeFrameRecordNumerationEnum	0..1	attr	This attribute defines the type of assigning freeze frame record numbers for event-specific freeze frame records.

Table 4.173: DiagnosticMemoryDestination

[constr_10421] Existence of attribute DiagnosticMemoryDestination.dtcStatusAvailabilityMask

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticMemoryDestination, the primitive attribute dtcStatusAvailabilityMask shall exist.⌋

[constr_10422] Existence of attribute DiagnosticMemoryDestination.eventDisplacementStrategy

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticMemoryDestination, attribute eventDisplacementStrategy shall exist.⌋

[constr_10423] Existence of attribute DiagnosticMemoryDestination.maxNumberOfEventEntries

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticMemoryDestination, attribute maxNumberOfEventEntries shall exist.⌋

Enumeration	**DiagnosticMemoryEntryStorageTriggerEnum**
Note	Trigger types to allocate an event memory entry.
Aggregated by	*DiagnosticMemoryDestination*.memoryEntryStorageTrigger
Literal	***Description***
confirmed	Status information of UDS DTC status bit 3 **Tags:** atp.EnumerationLiteralIndex=0
fdcThreshold	Threshold to allocate an event memory entry and to capture the Freeze Frame. **Tags:** atp.EnumerationLiteralIndex=1
testFailed	Status information of UDS DTC status bit 0. **Tags:** atp.EnumerationLiteralIndex=3

Table 4.174: DiagnosticMemoryEntryStorageTriggerEnum

Enumeration	**DiagnosticClearDtcLimitationEnum**
Note	Scope of the DEM_ClearDTC Api.
Aggregated by	*DiagnosticMemoryDestination*.clearDtcLimitation
Literal	***Description***
allSupportedDtcs	DEM_ClearDtc API accepts all supported DTC values. **Tags:** atp.EnumerationLiteralIndex=0
clearAllDtcs	DEM_ClearDtc API accepts ClearAllDTCs only. **Tags:** atp.EnumerationLiteralIndex=1

Table 4.175: DiagnosticClearDtcLimitationEnum

Enumeration	**DiagnosticEventDisplacementStrategyEnum**
Note	Defines the displacement strategy.
Aggregated by	*DiagnosticMemoryDestination*.eventDisplacementStrategy
Literal	***Description***
full	Event memory entry displacement is enabled, by consideration of priority active/passive status, and occurrence. **Tags:** atp.EnumerationLiteralIndex=0
none	Event memory entry displacement is disabled. **Tags:** atp.EnumerationLiteralIndex=1
prioOcc	Event memory entry displacement is enabled, by consideration of priority and occurrence (but without active/passive status). **Tags:** atp.EnumerationLiteralIndex=2

Table 4.176: DiagnosticEventDisplacementStrategyEnum

Enumeration	**DiagnosticStatusBitHandlingTestFailedSinceLastClearEnum**
Note	This enumeration controls whether the aging and displacement mechanism shall be applied to the 'TestFailedSinceLastClear' status bits.
Aggregated by	*DiagnosticMemoryDestination*.statusBitHandlingTestFailedSinceLastClear
Literal	***Description***
statusBitAgingAnd Displacement	The "TestFailedSinceLastClear" status bits are reset to 0, if aging or displacement applies. **Tags:** atp.EnumerationLiteralIndex=0
statusBitNormal	Aging and displacement has no impact on the "TestFailedSinceLastClear" status bits. **Tags:** atp.EnumerationLiteralIndex=1

Table 4.177: DiagnosticStatusBitHandlingTestFailedSinceLastClearEnum

Enumeration	**DiagnosticTypeOfFreezeFrameRecordNumerationEnum**
Note	FreezeFrame record numeration type
Aggregated by	*DiagnosticMemoryDestination*.typeOfFreezeFrameRecordNumeration
Literal	***Description***
calculated	Freeze frame records will be numbered consecutive starting by 1 in their chronological order. **Tags:** atp.EnumerationLiteralIndex=0
configured	Freeze frame records will be numbered based on the given configuration in their chronological order. **Tags:** atp.EnumerationLiteralIndex=1

Table 4.178: DiagnosticTypeOfFreezeFrameRecordNumerationEnum

Class	**DiagnosticMemoryDestinationPrimary**			
Note	This represents a primary memory for a diagnostic event. **Tags:** atp.recommendedPackage=DiagnosticMemoryDestinations			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemoryDestination*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
typeOfDtc Supported	DiagnosticTypeOfDtc SupportedEnum	0..1	attr	This attribute defines the format returned by Dem_Dcm GetTranslationType and does not relate to/influence the supported Dem functionality.

Table 4.179: DiagnosticMemoryDestinationPrimary

Class	**DiagnosticMemoryDestinationUserDefined**			
Note	This represents a user-defined memory for a diagnostic event. **Tags:** atp.recommendedPackage=DiagnosticMemoryDestinations			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemoryDestination*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
affectedBy ControlDTCSetting	Boolean	0..1	attr	This attribute configures how the content of the memory is affected by an active ControlDTCSetting or not: • If the attribute is set to **true**, the user-defined fault memory is **not** updated if ControlDTCSetting is off. • If the attribute is set to **false**, the user defined fault memory is updated even if ControlDTCSetting is off.
authenticationEnabled	DiagnosticAuthRoleProxy	0..1	aggr	The existence of this aggregation indicates that an authentication is foreseen. The details are clarified by the aggregated class. **Tags:** atp.Splitkey=authenticationEnabled
memoryId	PositiveInteger	0..1	attr	This represents the identifier of the user-defined memory.

Table 4.180: DiagnosticMemoryDestinationUserDefined

[constr_1379] Existence of DiagnosticMemoryDestinationPrimary

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈Within the scope of one DiagnosticContributionSet only one DiagnosticMemoryDestinationPrimary shall exist.⌋

4.4.5 Diagnostic Extended Data Record

[TPS_DEXT_03008] Semantics of meta-class DiagnosticExtendedDataRecord ⌈A DiagnosticExtendedDataRecord contains DiagnosticExtendedDataRecordElement in an ordered collection.

A DiagnosticExtendedDataRecord.element represents **either**

- a primitive data object **or**
- a fixed-size array of primitive data objects, in which case the number of elements in the array is determined by attribute DiagnosticExtendedDataRecordElement.numberOfElements.

The definition of DiagnosticExtendedDataRecord is **independent of the underlying diagnostic protocol**.

⌋

[constr_1355] Value of extendedDataRecord.recordNumber

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈To be compliant with the ISO standard, the value of extendedDataRecord.recordNumber shall be set in the interval as defined in [17, ISO 14229-1].⌋

[constr_1509] Value of extendedDataRecord.recordNumber shall be unique within primary fault memory

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For all DiagnosticTroubleCodeProps that refer to DiagnosticMemoryDestinationPrimary in the role diagnosticMemory there shall be no two extendedDataRecord.recordNumber with the same value.⌋

[constr_1511] Value of extendedDataRecord.recordNumber shall be unique within user-defined fault memory

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For all DiagnosticTroubleCodeProps that refer to DiagnosticMemoryDestinationUserDefined in the role diagnosticMemory there shall be no two extendedDataRecord.recordNumber with the same value for any DiagnosticMemoryDestinationUserDefined referenced as DiagnosticTroubleCodeProps.diagnosticMemory with a given value of memoryId.⌋

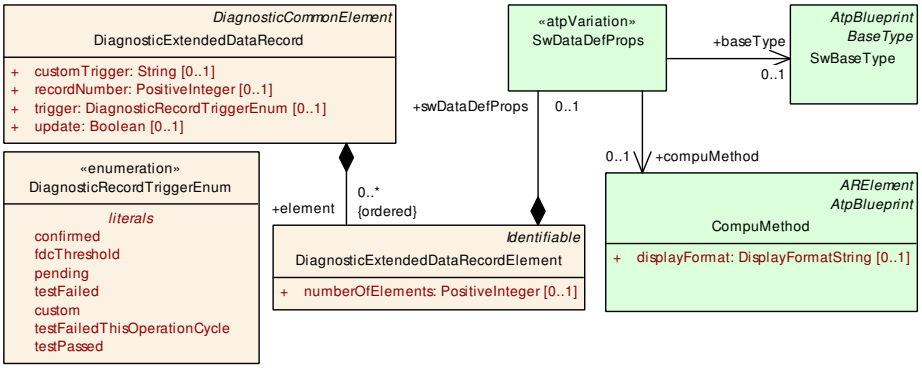

Figure 4.62: Modeling of DiagnosticExtendedDataRecord

[constr_10676] Allowed attributes of DiagnosticExtendedDataRecordElement.swDataDefProps

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈The only allowed attributes of SwDataDefProps aggregated in the role DiagnosticExtendedDataRecordElement.swDataDefProps are

- compuMethod for the definition of a conversion method from internal to physical representation in the diagnostic client
- baseType for the definition of the memory layout of the DiagnosticExtendedDataRecord.

None of the other attributes of SwDataDefProps shall be used in the context of DiagnosticExtendedDataRecordElement.swDataDefProps.⌋

[constr_10677] Allowed values for attribute DiagnosticExtendedDataRecordElement.swDataDefProps.baseType.baseTypeSize

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈The only allowed values filled into attribute DiagnosticExtendedDataRecordElement.swDataDefProps.baseType.baseTypeSize shall be a multiple of 8.⌋

[TPS_DEXT_01143] Definition of a custom trigger for an extended data record ⌈It is possible to define a custom trigger for the capturing of an extended data record. The custom nature, however, severely limits the possibilities for a formal description of such a trigger. Therefore, AUTOSAR can only provide a means to verbally describe how the custom trigger shall operate.

For this purpose, the attribute DiagnosticExtendedDataRecord.customTrigger has been defined.⌋

[constr_1616] Existence of attribute DiagnosticExtendedDataRecord.customTrigger

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈The attribute DiagnosticExtendedDataRecord.customTrigger shall only exist if the attribute DiagnosticExtendedDataRecord.trigger is set to the value DiagnosticRecordTriggerEnum.custom.⌋

Class	**DiagnosticExtendedDataRecord**			
Note	Description of an extended data record. **Tags:** atp.recommendedPackage=DiagnosticExtendedDataRecords			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
customTrigger	String	0..1	attr	This attribute shall be taken to verbally describe the nature of the custom trigger.
element (ordered)	DiagnosticExtendedDataRecordElement	*	aggr	This aggregation represents the collection of elements of the enclosing Extentded Data Record.
recordNumber	PositiveInteger	0..1	attr	This attribute specifies an unique identifier for an extended data record.
trigger	DiagnosticRecordTriggerEnum	0..1	attr	This attribute specifies the primary trigger to allocate an event memory entry.
update	Boolean	0..1	attr	This attribute defines when an extended data record is captured. true: This extended data record is captured every time. false: This extended data record is only captured for new event memory entries.

Table 4.181: DiagnosticExtendedDataRecord

Class	**DiagnosticExtendedDataRecordElement**			
Note	This meta-class represents an element of an extended data record. Such an element represents a primitive data object or a fixed-size array of primitive data objects without further internal structure (no structure allowed).			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticExtendedDataRecord.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
numberOf Elements	PositiveInteger	0..1	attr	The existence of this attribute turns the data instance into a fixed-size array of data. The attribute determines the size of the array in terms of how many elements the array shall take. The aggregation of SwDataDefProps (especially with respect to SwBaseType) applies specifically to an element of the array (as a representation of all array elements) instead of the entire array itself!
swDataDef Props	SwDataDefProps	0..1	aggr	This aggregation allows for the specification of various properties for the enclosing element. **Tags:** atp.Splitkey=swDataDefProps

Table 4.182: DiagnosticExtendedDataRecordElement

[constr_1756] Existence of attributes `DiagnosticExtendedDataRecord.trigger` and `update`

Imposition time: CP: `IT_DiagDes`, AP: `IT_DiagDes`

⌈For each `DiagnosticExtendedDataRecord`, attributes `trigger` and `update` shall only exist if at least one `DiagnosticExtendedDataRecordElement` is aggregated by a `DiagnosticExtendedDataRecord` in the role `element` that is referenced in either of the roles

- `DiagnosticEdrSenderPortMapping.recordElement`
- `DiagnosticEdrServerPortMapping.recordElement`

⌋

Please note that [constr_1756] is the equivalent of [constr_10732], see [7, TPS Manifest Specification] Chapter 5.2.2.10.

[constr_1760] Existence of `DiagnosticExtendedDataRecord.element`

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈For each `DiagnosticExtendedDataRecord`, at least one aggregation of `DiagnosticExtendedDataRecordElement` in the role `element` shall exist.⌋

[constr_1832] Existence of `DiagnosticExtendedDataRecord.recordNumber`

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈For each `DiagnosticExtendedDataRecord`, attribute `recordNumber` shall exist.⌋

[constr_1859] Usage of DiagnosticRecordTriggerEnum.testFailedThis-OperationCycle

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈The enumeration value DiagnosticRecordTriggerEnum.testFailedThis-OperationCycle shall only be used in the context of meta-class Diagnostic-FreezeFrame.⌋

[constr_10364] Usage of DiagnosticRecordTriggerEnum.testPassed

Imposition time: CP: IT_DiagDes

⌈The enumeration value DiagnosticRecordTriggerEnum.testPassed shall only be used in the context of DiagnosticExtendedDataRecord.trigger.⌋

The meaning of [constr_10364] is that the use of DiagnosticRecordTrigger-Enum.testPassed in a DiagnosticFreezeFrame is explicitly not allowed.

Enumeration	**DiagnosticRecordTriggerEnum**
Note	Triggers to allocate an event memory entry.
Aggregated by	DiagnosticExtendedDataRecord.trigger, DiagnosticFreezeFrame.trigger
Literal	***Description***
confirmed	capture on "Confirmed" **Tags:** atp.EnumerationLiteralIndex=0
custom	implement custom capture **Tags:** atp.EnumerationLiteralIndex=4
fdcThreshold	capture on "FDC Threshold" **Tags:** atp.EnumerationLiteralIndex=1
pending	capture on "Pending" **Tags:** atp.EnumerationLiteralIndex=2
testFailed	capture on "Test Failed" **Tags:** atp.EnumerationLiteralIndex=3
testFailedThis OperationCycle	Test Failed This Operation Cycle. **Tags:** atp.EnumerationLiteralIndex=5 This EnumerationLiteral is only used by the AUTOSAR Classic Platform.
testPassed	Capture on testFailed bit transition 1 --> 0. **Tags:** atp.EnumerationLiteralIndex=6 This EnumerationLiteral is only used by the AUTOSAR Classic Platform.

Table 4.183: DiagnosticRecordTriggerEnum

4.4.6 Diagnostic Freeze Frame

[TPS_DEXT_03009] Semantics of DiagnosticFreezeFrame ⌈A Diagnostic-FreezeFrame needs an ordered list of references to DiagnosticData-Identifiers. However, these references are not modeled directly but in the context of meta-class DiagnosticTroubleCodeProps.⌋

For more details, please refer to Figure 4.58.

[constr_1357] Value of `freezeFrame.recordNumber`

Imposition time: CP: `IT_DiagDes`

⌈To be compliant with ISO, the value of `freezeFrame.recordNumber` shall be set in the interval as defined in [17, ISO 14229-1].⌋

[constr_1512] `freezeFrame.recordNumber` shall be unique within primary fault memory

Imposition time: CP: `IT_DiagDes`

⌈For all `DiagnosticTroubleCodeProps` that refer to `DiagnosticMemoryDestinationPrimary` in the role `diagnosticMemory` there shall be no two `freezeFrame.recordNumber` with the same value.⌋

[constr_1514] `freezeFrame.recordNumber` shall be unique within user-defined fault memory

Imposition time: CP: `IT_DiagDes`

⌈For all `DiagnosticTroubleCodeProps` that refer to `DiagnosticMemoryDestinationUserDefined` in the role `diagnosticMemory` there shall be no two `freezeFrame.recordNumber` with the same value for any `DiagnosticMemoryDestinationUserDefined` referenced as `DiagnosticTroubleCodeProps.diagnosticMemory` with a given value of `memoryId`.⌋

Class	**DiagnosticFreezeFrame**			
Note	This element describes combinations of DIDs for a non OBD relevant freeze frame. **Tags:** atp.recommendedPackage=DiagnosticFreezeFrames			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
customTrigger	String	0..1	attr	This attribute shall be taken to verbally describe the nature of the custom trigger.
recordNumber	PositiveInteger	0..1	attr	This attribute defines a record number for a freeze frame record. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
trigger	DiagnosticRecordTriggerEnum	0..1	attr	This attribute defines the primary trigger to allocate an event memory entry.
update	Boolean	0..1	attr	This attribute defines the approach when the freeze frame record is stored/updated. true: FreezeFrame record is captured every time. false: FreezeFrame record is only captured for new event memory entries.

Table 4.184: DiagnosticFreezeFrame

[TPS_DEXT_01144] Definition of a custom trigger for a freeze frame ⌈It is possible to define a custom trigger for the capturing of a freeze frame. The custom nature, however, severely limits the possibilities for a formal description of such a trigger. Therefore, AUTOSAR can only provide a means to verbally describe how the custom trigger shall operate.

For this purpose, the attribute DiagnosticFreezeFrame.customTrigger has been defined.⌋

[constr_1833] Existence of DiagnosticFreezeFrame.trigger

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticFreezeFrame, attribute trigger shall exist.⌋

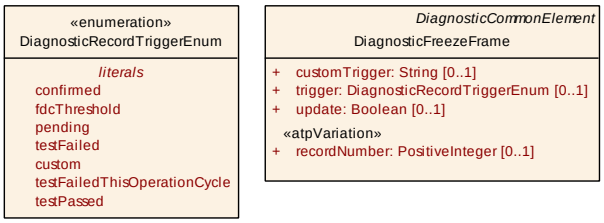

Figure 4.63: Modeling of DiagnosticFreezeFrame

[constr_1617] Existence of attribute DiagnosticFreezeFrame.customTrigger

Imposition time: CP: IT_DiagDes

⌈The attribute DiagnosticFreezeFrame.customTrigger shall only exist if the attribute DiagnosticFreezeFrame.trigger is set to the value DiagnosticRecordTriggerEnum.custom.⌋

4.4.7 Diagnostic Condition

[TPS_DEXT_03010] Combination of DiagnosticConditions to DiagnosticConditionGroups ⌈DiagnosticConditions are combined into DiagnosticConditionGroups and define a certain number of checks (e.g. a correct voltage range) before the event report is accepted or the event gets qualified.⌋

[TPS_DEXT_03001] Consideration of different types of conditions ⌈There are two different types of conditions: DiagnosticEnableConditions and DiagnosticStorageConditions:

- As long as the DiagnosticEnableCondition is not fulfilled, the event reports are not valid and therefore will not be accepted.
- As long as the DiagnosticStorageCondition is not fulfilled, the event is not stored in the event memory.

⌋

Class	**DiagnosticCondition** (abstract)			
Note	Abstract element for StorageConditions and EnableConditions.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticEnableCondition, DiagnosticStorageCondition			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
initValue	Boolean	0..1	attr	Defines the initial status for enable or disable of acceptance/storage of event reports of a diagnostic event. The value is the initialization after power up (before this condition is reported the first time). true: acceptance/storage of a diagnostic event enabled false: acceptance/storage of a diagnostic event disabled

Table 4.185: DiagnosticCondition

[constr_1834] Existence of DiagnosticCondition.initValue

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticCondition, attribute initValue shall exist.⌋

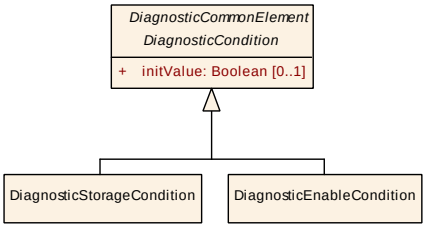

Figure 4.64: Modeling of DiagnosticCondition

Class	**DiagnosticEnableCondition**			
Note	Specification of an enable condition. **Tags:** atp.recommendedPackage=DiagnosticConditions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticCondition*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.186: DiagnosticEnableCondition

Class	**DiagnosticStorageCondition**			
Note	Specification of a storage condition. **Tags:** atp.recommendedPackage=DiagnosticConditions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticCondition*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.187: DiagnosticStorageCondition

4.4.8 Diagnostic Debouncing

[TPS_DEXT_01048] Actual algorithm for the diagnostic event debouncing ⌈The actual algorithm for the debouncing is represented by subclasses of DiagEventDebounceAlgorithm aggregated in the role DiagnosticDebounceAlgorithmProps.debounceAlgorithm.⌋

In other words, the debouncing of diagnostic events can be formulated in three different ways:

- The DiagEventDebounceCounterBased represents the ability to implement a counter-based debouncing.
- The DiagEventDebounceTimeBased represents the ability to implement a time-based debouncing.
- The DiagEventDebounceMonitorInternal represents the ability to implement the debouncing inside the implementation of the diagnostic monitor itself.

[constr_1359] Condition for the existence of attribute DiagnosticDebounceAlgorithmProps.debounceCounterStorage

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈Attribute debounceCounterStorage of meta-class DiagnosticDebounceAlgorithmProps shall only exist if the aggregation of attribute debounceAlgorithm at DiagnosticDebounceAlgorithmProps actually aggregates a DiagEventDebounceCounterBased.⌋

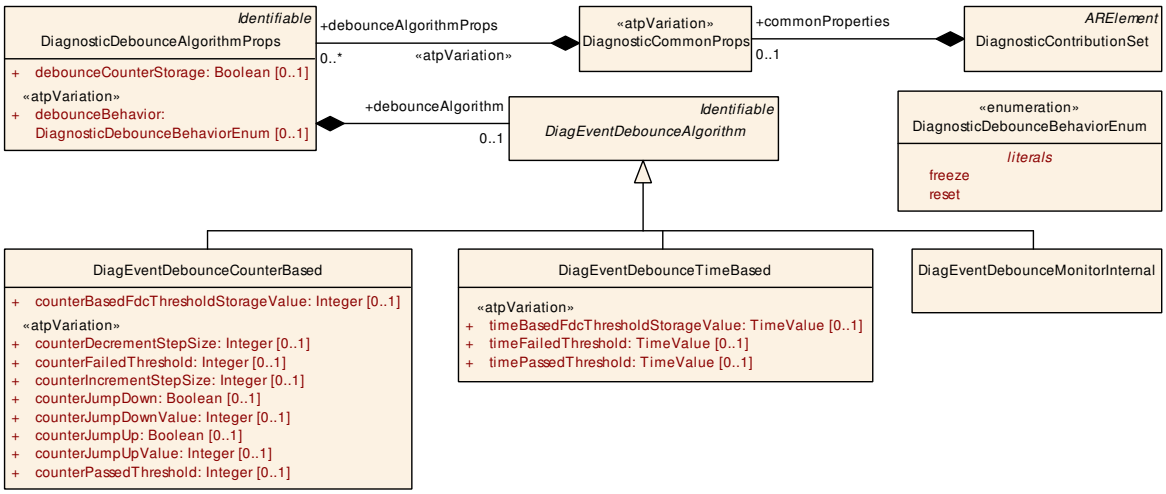

Figure 4.65: Details of DiagnosticDebounceAlgorithmProps

Class	DiagnosticDebounceAlgorithmProps			
Note	Defines properties for the debounce algorithm class.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticCommonProps.debounceAlgorithmProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
debounce Algorithm	DiagEventDebounce Algorithm	0..1	aggr	This represents the actual debounce algorithm. **Tags:** atp.Splitkey=debounceAlgorithm, debounce Algorithm.shortName
debounce Behavior	DiagnosticDebounce BehaviorEnum	0..1	attr	This attribute defines how the event debounce algorithm will behave, if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
debounce CounterStorage	Boolean	0..1	attr	Switch to store the debounce counter value non-volatile or not. true: debounce counter value shall be stored non-volatile false: debounce counter value is volatile Please note that this attribute is not relevant for the adaptive platform. This Attribute is only used by the AUTOSAR Classic Platform.

Table 4.188: DiagnosticDebounceAlgorithmProps

[constr_10418] Existence of attribute `DiagnosticDebounceAlgorithmProps.debounceAlgorithm`

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈For each `DiagnosticDebounceAlgorithmProps`, the primitive attribute `debounceAlgorithm` shall exist.⌋

Class	***DiagEventDebounceAlgorithm*** (abstract)			
Note	This class represents the ability to specify the pre-debounce algorithm which is selected and/or required by the particular monitor. This class inherits from Identifiable in order to allow further documentation of the expected or implemented debouncing and to use the category for the identification of the expected / implemented debouncing.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	DiagEventDebounceCounterBased, DiagEventDebounceMonitorInternal, DiagEventDebounceTime Based			
Aggregated by	DiagnosticDebounceAlgorithmProps.debounceAlgorithm, DiagnosticEventNeeds.diagEventDebounce Algorithm			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.189: DiagEventDebounceAlgorithm

Class	**DiagEventDebounceCounterBased**			
Note	This meta-class represents the ability to indicate that the counter-based debounce algorithm shall be used by the DEM for this diagnostic monitor. This is related to set the ECUC choice container DemDebounceAlgorithmClass to DemDebounce CounterBased.			
Base	*ARObject*, *DiagEventDebounceAlgorithm*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticDebounceAlgorithmProps.debounceAlgorithm, DiagnosticEventNeeds.diagEventDebounce Algorithm			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
counterBased FdcThreshold StorageValue	Integer	0..1	attr	Threshold to allocate an event memory entry and to capture the Freeze Frame.
counter DecrementStep Size	Integer	0..1	attr	This value shall be taken to decrement the internal debounce counter. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
counterFailed Threshold	Integer	0..1	attr	This value defines the event-specific limit that indicates the "failed" counter status. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
counter IncrementStep Size	Integer	0..1	attr	This value shall be taken to increment the internal debounce counter. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
counterJump Down	Boolean	0..1	attr	This value activates or deactivates the counter jump-down behavior. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
counterJump DownValue	Integer	0..1	attr	This value represents the initial value of the internal debounce counter if the counting direction changes from incrementing to decrementing. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
counterJumpUp	Boolean	0..1	attr	This value activates or deactivates the counter jump-up behavior. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
counterJumpUp Value	Integer	0..1	attr	This value represents the initial value of the internal debounce counter if the counting direction changes from decrementing to incrementing. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
counterPassed Threshold	Integer	0..1	attr	This value defines the event-specific limit that indicates the "passed" counter status. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table 4.190: DiagEventDebounceCounterBased

[constr_1835] Existence of the attribute `DiagEventDebounceCounterBased.counterDecrementStepSize`

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈For each `DiagEventDebounceCounterBased`, attribute `counterDecrementStepSize` shall exist.⌋

[constr_1836] Existence of the attribute DiagEventDebounceCounterBased.counterIncrementStepSize

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagEventDebounceCounterBased, attribute counterIncrementStepSize shall exist.⌋

[constr_1837] Existence of the attribute DiagEventDebounceCounterBased.counterFailedThreshold

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagEventDebounceCounterBased, attribute counterFailedThreshold shall exist.⌋

[constr_1838] Existence of the attribute DiagEventDebounceCounterBased.counterPassedThreshold

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagEventDebounceCounterBased, attribute counterPassedThreshold shall exist.⌋

[constr_1766] Existence of the attribute DiagEventDebounceCounterBased.counterJumpDownValue

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagEventDebounceCounterBased, attribute counterJumpDownValue shall only exist if attribute counterJumpDown exists and is set to true.⌋

[constr_1767] Existence of the attribute DiagEventDebounceCounterBased.counterJumpUpValue

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagEventDebounceCounterBased, attribute counterJumpUpValue shall only exist if attribute counterJumpUp exists and is set to true.⌋

Class	**DiagEventDebounceTimeBased**			
Note	This meta-class represents the ability to indicate that the time-based pre-debounce algorithm shall be used by the Dem for this diagnostic monitor. This is related to set the EcuC choice container DemDebounceAlgorithmClass to DemDebounceTime Base.			
Base	*ARObject*, *DiagEventDebounceAlgorithm*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticDebounceAlgorithmProps.debounceAlgorithm, DiagnosticEventNeeds.diagEventDebounceAlgorithm			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
timeBasedFdcThresholdStorageValue	TimeValue	0..1	attr	Threshold to allocate an event memory entry and to capture the Freeze Frame. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

▽

△

Class	DiagEventDebounceTimeBased			
timeFailed Threshold	TimeValue	0..1	attr	This value represents the event-specific delay indicating the "failed" status. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
timePassed Threshold	TimeValue	0..1	attr	This value represents the event-specific delay indicating the "passed" status. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table 4.191: DiagEventDebounceTimeBased

[constr_1839] Existence of the attribute DiagEventDebounceTimeBased.timeFailedThreshold

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagEventDebounceTimeBased, attribute timeFailedThreshold shall exist.⌋

[constr_1840] Existence of attribute DiagEventDebounceTimeBased.timePassedThreshold

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagEventDebounceTimeBased, attribute timePassedThreshold shall exist.⌋

Class	DiagEventDebounceMonitorInternal			
Note	This meta-class represents the ability to indicate that no Dem pre-debounce algorithm shall be used for this diagnostic monitor. The SWC might implement an internal debouncing algorithm and report qualified (debounced) results to the Dem/DM.			
Base	*ARObject*, *DiagEventDebounceAlgorithm*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticDebounceAlgorithmProps.debounceAlgorithm, DiagnosticEventNeeds.diagEventDebounceAlgorithm			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 4.192: DiagEventDebounceMonitorInternal

Enumeration	DiagnosticDebounceBehaviorEnum
Note	Event debounce algorithm behavior options.
Aggregated by	DiagnosticDebounceAlgorithmProps.debounceBehavior
Literal	**Description**
freeze	The event debounce counter will be frozen with the current value and will not change while a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. After all related enable conditions are fulfilled and ControlDTCSetting of the related event is enabled again, the event qualification will continue with the next report of the event (i.e. SetEventStatus). **Tags:** atp.EnumerationLiteralIndex=0

▽

△

Enumeration	**DiagnosticDebounceBehaviorEnum**
reset	The event debounce counter will be reset to initial value if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. The qualification of the event will be restarted with the next valid event report. **Tags:** atp.EnumerationLiteralIndex=1

Table 4.193: DiagnosticDebounceBehaviorEnum

4.4.9 Diagnostic Condition Group

[TPS_DEXT_01084] Semantics of meta-class `DiagnosticConditionGroup`
⌈`DiagnosticConditionGroup`s are used to collect `DiagnosticCondition`s that in turn are assigned to `DiagnosticEvent`s.⌋

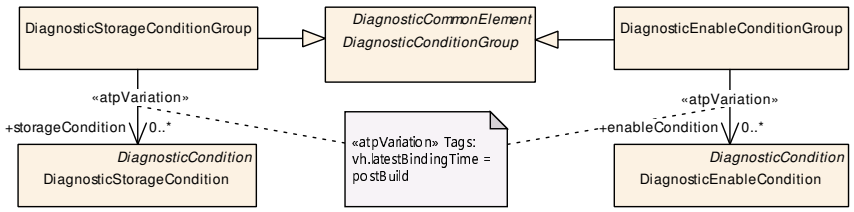

Figure 4.66: Modeling of DiagnosticConditionGroup

Class	***DiagnosticConditionGroup*** (abstract)			
Note	Abstract element for StorageConditionGroups and EnableConditionGroups.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticEnableConditionGroup, DiagnosticStorageConditionGroup			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.194: DiagnosticConditionGroup

Class	**DiagnosticEnableConditionGroup**			
Note	Enable condition group which includes one or several enable conditions. **Tags:** atp.recommendedPackage=DiagnosticConditions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticConditionGroup*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**DiagnosticEnableConditionGroup**			
enableCondition	DiagnosticEnable Condition	*	ref	Reference to enableConditions that are part of the Enable ConditionGroup. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=enableCondition, enableCondition.diagnostic EnableCondition, enableCondition.variationPoint.short Label vh.latestBindingTime=postBuild

Table 4.195: DiagnosticEnableConditionGroup

[constr_1841] Existence of attribute DiagnosticEnableConditionGroup.enableCondition

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticEnableConditionGroup, the attribute enableCondition shall exist.⌋

Class	**DiagnosticStorageConditionGroup**			
Note	Storage condition group which includes one or several storage conditions. **Tags:** atp.recommendedPackage=DiagnosticConditions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticConditionGroup*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
storage Condition	DiagnosticStorage Condition	*	ref	Reference to storageConditions that are part of the StorageConditionGroup. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=storageCondition, storage Condition.diagnosticStorageCondition, storage Condition.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 4.196: DiagnosticStorageConditionGroup

[constr_1842] Existence of DiagnosticStorageConditionGroup.storage-Condition

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticStorageConditionGroup, attribute storageCondition shall exist.⌋

4.4.10 Diagnostic Operation Cycle

[TPS_DEXT_01087] Semantics of DiagnosticOperationCycle ⌈Different types of DiagnosticOperationCycles are supported and defined by the type attribute, e.g. the time between ignition on and ignition off.⌋

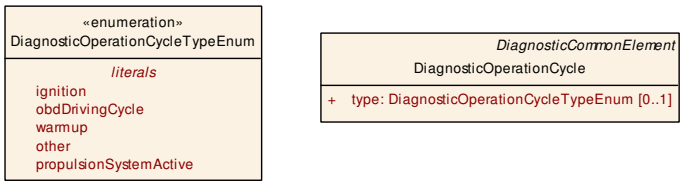

Figure 4.67: Modeling of DiagnosticOperationCycle

Class	**DiagnosticOperationCycle**			
Note	Definition of an operation cycle that is the base of the event qualifying and for Dem scheduling. **Tags:** atp.recommendedPackage=DiagnosticOperationCycles			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
type	DiagnosticOperationCycleTypeEnum	0..1	attr	Operation cycles types for the Dem.

Table 4.197: DiagnosticOperationCycle

Enumeration	**DiagnosticOperationCycleTypeEnum**
Note	Operation cycles types used to identify certain Operation cycles with a certain semantics.
Aggregated by	DiagnosticOperationCycle.type
Literal	***Description***
ignition	Ignition ON / OFF cycle **Tags:** atp.EnumerationLiteralIndex=0
obdDrivingCycle	OBD Driving cycle **Tags:** atp.EnumerationLiteralIndex=1
other	further operation cycle **Tags:** atp.EnumerationLiteralIndex=2
propulsionSystemActive	Zero Emission Vehicle Propulsion System Active operation cycle **Tags:** atp.EnumerationLiteralIndex=6
warmup	OBD Warm up cycle **Tags:** atp.EnumerationLiteralIndex=5

Table 4.198: DiagnosticOperationCycleTypeEnum

4.4.11 Diagnostic Aging

[TPS_DEXT_03021] Aging ⌈It is possible to remove a specific event from the event memory, if its fault conditions are not fulfilled for a certain period of time. This process is called as aging or unlearning.

This semantics is formalized by means of the meta-class `DiagnosticAging`.⌋

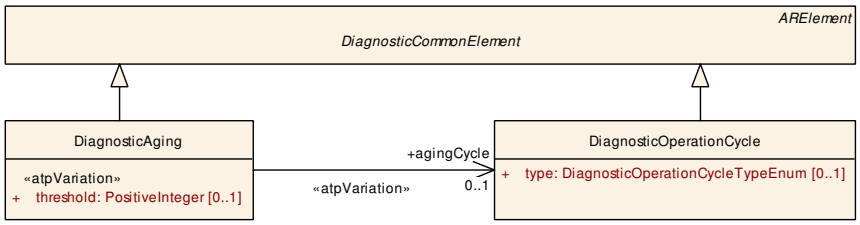

Figure 4.68: Modeling of `DiagnosticAging`

Class	**DiagnosticAging**			
Note	Defines the aging algorithm. **Tags:** atp.recommendedPackage=DiagnosticAgings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
agingCycle	DiagnosticOperation Cycle	0..1	ref	This represents the applicable aging cycle. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=agingCycle, agingCycle.diagnosticOperation Cycle, agingCycle.variationPoint.shortLabel vh.latestBindingTime=postBuild
threshold	PositiveInteger	0..1	attr	Number of aging cycles needed to unlearn/delete the event. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table 4.199: DiagnosticAging

[constr_1848] Existence of attribute `DiagnosticAging.agingCycle`

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈For each `DiagnosticAging`, attribute `agingCycle` shall exist.⌋

[constr_1849] Existence of attribute `DiagnosticAging.threshold`

Imposition time: AP: `IT_DiagDes`, CP: `IT_DiagDes`

⌈For each `DiagnosticAging`, attribute `threshold` shall exist.⌋

4.4.12 Diagnostic Indicator

[TPS_DEXT_03022] Different kinds of `DiagnosticIndicator`s ⌈Different `type`s of Indicators can be defined with the `DiagnosticIndicator` element. For this, the attribute `DiagnosticIndicator.type` shall be used.⌋

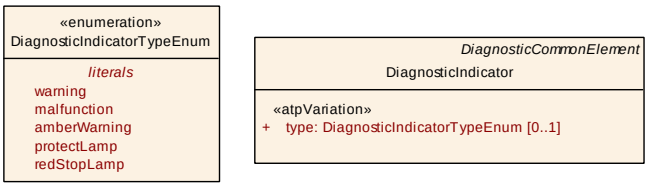

Figure 4.69: Modeling of `DiagnosticIndicator`

Class	**DiagnosticIndicator**			
Note	Definition of an indicator. **Tags:** atp.recommendedPackage=DiagnosticIndicators			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
type	DiagnosticIndicatorTypeEnum	0..1	attr	Defines the type of the indicator. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table 4.200: DiagnosticIndicator

Enumeration	**DiagnosticIndicatorTypeEnum**
Note	Type of an indicator.
Aggregated by	DiagnosticIndicator.type, IndicatorStatusNeeds.type
Literal	***Description***
amberWarning	Amber Warning Lamp **Tags:** atp.EnumerationLiteralIndex=0
malfunction	Malfunction Indicator Lamp **Tags:** atp.EnumerationLiteralIndex=1
protectLamp	Protect Lamp **Tags:** atp.EnumerationLiteralIndex=2
redStopLamp	Red Stop Lamp **Tags:** atp.EnumerationLiteralIndex=3
warning	Warning **Tags:** atp.EnumerationLiteralIndex=4

Table 4.201: DiagnosticIndicatorTypeEnum

4.4.13 Diagnostic Test Result

The meta-class `DiagnosticTestResult` allows for a formal definition of a diagnostic test result.

The purpose of this meta-class is to support the reporting of the latest test results back to a client. This is of special importance for the `OBD` service mode 0x06, see Chapter 4.3.9.5.

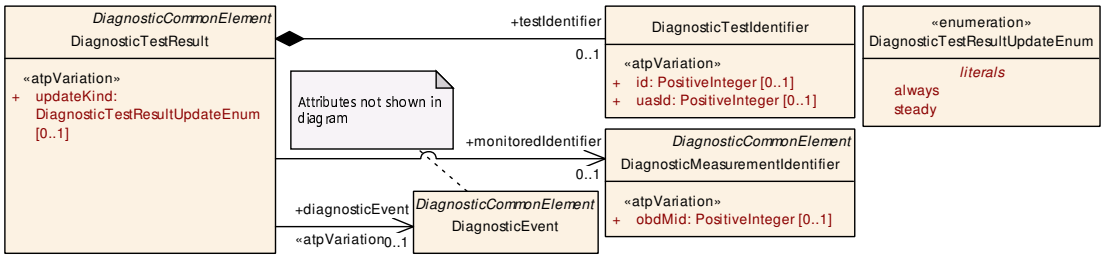

Figure 4.70: Modeling of `DiagnosticTestResult`

Class	**DiagnosticTestResult**			
Note	This meta-class represents the ability to define diagnostic test results. **Tags:** atp.recommendedPackage=DiagnosticTestResults			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnosticEvent	DiagnosticEvent	0..1	ref	This attribute represents the diagnostic event that is related to the diagnostic test result. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=diagnosticEvent, diagnosticEvent.diagnosticEvent, diagnosticEvent.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
monitored Identifier	Diagnostic MeasurementIdentifier	0..1	ref	This attribute represents the related diagnostic monitored identifier. **Tags:** atp.Splitkey=monitoredIdentifier
testIdentifier	DiagnosticTestIdentifier	0..1	aggr	This attribute represents the applicable test identifier. **Tags:** atp.Splitkey=testIdentifier
updateKind	DiagnosticTestResult UpdateEnum	0..1	attr	This attribute controls the update behavior of the enclosing DiagnosticTestResult. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table 4.202: DiagnosticTestResult

[constr_1850] Existence of the aggregation in the role `DiagnosticTestResult.testIdentifier`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticTestResult`, the aggregation of meta-class `DiagnosticTestIdentifier` in the role `testIdentifier` shall exist.⌋

[constr_1851] Existence of the reference `DiagnosticTestResult.monitoredIdentifier`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticTestResult`, the reference to meta-class `DiagnosticTestIdentifier` in the role `monitoredIdentifier` shall exist.⌋

Enumeration	**DiagnosticTestResultUpdateEnum**
Note	This meta-class represents the ability to define the update behavior of a DiagnosticTestResult.
Aggregated by	DiagnosticTestResult.updateKind
Literal	***Description***
always	Any DTR result reported by the monitor is used by the Dem. **Tags:** atp.EnumerationLiteralIndex=0
steady	The Dem accepts reported DTRs only when the configured debouncing mechanism is stable at the FAIL or PASS limit. **Tags:** atp.EnumerationLiteralIndex=1

Table 4.203: DiagnosticTestResultUpdateEnum

Class	**DiagnosticTestIdentifier**			
Note	This meta-class represents the ability to create a diagnostic test identifier.			
Base	*ARObject*			
Aggregated by	DiagnosticTestResult.testIdentifier			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
id	PositiveInteger	0..1	attr	This represents the numerical id associated with the diagnostic test identifier. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
uasId	PositiveInteger	0..1	attr	This represents the unit and scaling Id of the diagnostic test result. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table 4.204: DiagnosticTestIdentifier

[constr_10412] Existence of attribute `DiagnosticTestIdentifier.id`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticTestIdentifier`, attribute `id` shall exist.⌋

[constr_10413] Existence of attribute `DiagnosticTestIdentifier.uasId`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticTestIdentifier`, attribute `uasId` shall exist.⌋

Class	**DiagnosticMeasurementIdentifier**			
Note	This meta-class represents the ability to describe a measurement identifier. **Tags:** atp.recommendedPackage=DiagnosticMeasurementIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
obdMid	PositiveInteger	0..1	attr	This represents the numerical measurement Id **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table 4.205: DiagnosticMeasurementIdentifier

[constr_10414] Existence of attribute `DiagnosticMeasurementIdentifier.obdMid`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticMeasurementIdentifier`, attribute `obdMid` shall exist.⌋

Please note that a very similar modeling exists in the form of `ObdMonitorServiceNeeds.updateKind` in the context of the [8, TPS Software Component Template].

The implementer of the software-component is aware of the way the software-component uses debouncing and how it calls the Dem.

The implementer of the software-component has all the information needed to set a proper value for the attribute `ObdMonitorServiceNeeds.updateKind`.

The author of the DEXT model, on the other hand, might notice that a diagnostics software-component is broken, and might use the DEXT model as a work-around to fix a wrong setting of `ObdMonitorServiceNeeds.updateKind` by means of `DiagnosticTestResult.updateKind`.

4.4.14 OBD-related aspects of Dem Configuration

The support for OBD-related (see [19, ISO 15031-5]) modeling requires the addition of some pretty global (i.e. on the level of an entire ECU) attributes that could be added to the `EcuInstance`.

However, this would not align with the idea of a decentralized configuration of the diagnostic stack where information is added at a point in time when an actual `EcuInstance` is not yet available.

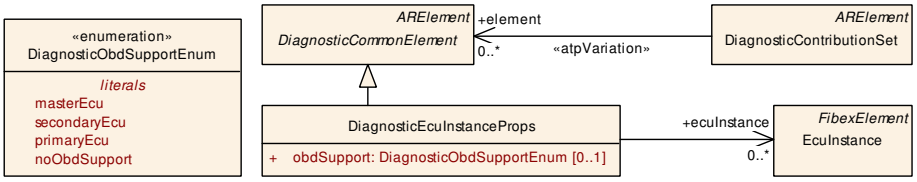

Figure 4.71: Modeling of `DiagnosticEcuInstanceProps`

Therefore, the attributes applying for the ECU-level are added to the meta-class `DiagnosticEcuInstanceProps`.

[TPS_DEXT_01122] Indication whether a `EcuInstance` supports OBD ⌈The attribute `DiagnosticEcuInstanceProps.obdSupport` is taken to define whether a given `EcuInstance` shall support `OBD` and in which way `OBD` shall be supported on this Ecu.⌋

Class	**DiagnosticEcuInstanceProps**			
Note	This meta-class represents the ability to model properties that are specific for a given EcuInstance but on the other hand represent purely diagnostic-related information. In the spirit of decentralized configuration it is therefore possible to specify the diagnostic-related information related to a given EcuInstance even if the EcuInstance does not yet exist. **Tags:** atp.recommendedPackage=DiagnosticEcuInstancePropss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ecuInstance	EcuInstance	*	ref	This represents the actual EcuInstance to which the information contained in the DiagnosticEcuInstance contribute. **Tags:** atp.Splitkey=ecuInstance
obdSupport	DiagnosticObdSupportEnum	0..1	attr	This attribute is used to specify the role (if applicable) in which the DiagnosticEcuInstance supports OBD.

Table 4.206: DiagnosticEcuInstanceProps

[constr_1852] Existence of attribute DiagnosticEcuInstanceProps.obdSupport

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEcuInstanceProps, attribute obdSupport shall exist.⌋

Enumeration	**DiagnosticObdSupportEnum**
Note	This meta-class represents the ability to model the roles in which a participation in OBD is foreseen. At the moment, this applies exclusively to the Dem. However, future extension of the Dcm may require this setting as well.
Aggregated by	DiagnosticEcuInstanceProps.obdSupport
Literal	***Description***
masterEcu	This represent the role "master ECU". **Tags:** atp.EnumerationLiteralIndex=0
noObdSupport	This represents the ability to explicitly specify that no participation in OBD is foreseen. **Tags:** atp.EnumerationLiteralIndex=1
primaryEcu	This represents the role "primary ECU". **Tags:** atp.EnumerationLiteralIndex=2
secondaryEcu	This represents the role "secondary ECU". **Tags:** atp.EnumerationLiteralIndex=3

Table 4.207: DiagnosticObdSupportEnum

Another OBD-related feature of the Dem is the support for the computation of the *In-Use-Monitor Performance Ratio* (IUMPR). It is possible to define a so-called DiagnosticIumprGroup that encapsulates the computation of a group of DiagnosticEvents.

The modeling of the DiagnosticIumprDenominatorGroup supports the configuration of additional environmental conditions that shall be applied to the increment of the IUMPR denominator.

From the modeling point of view, neither DiagnosticIumprGroup nor DiagnosticIumprDenominatorGroup reference the associated DiagnosticEvent directly.

But since the DiagnosticIumpr referenced in the role iumpr in turn references at most one DiagnosticEvent, the resulting relation effectively boils down to DiagnosticIumprGroup and DiagnosticIumprDenominatorGroup being able to precisely define the collection of affected DiagnosticEvents.

[TPS_DEXT_01148] Standardized values of DiagnosticIumprGroup.category
⌈The semantics of meta-class DiagnosticIumprGroup is identified by means of the value of attribute DiagnosticIumprGroup.category.

Standardized values of attribute DiagnosticIumprGroup.category are:

- IUMPR_AFRI1
- IUMPR_AFRI2
- IUMPR_BOOSTPRS
- IUMPR_CAT1
- IUMPR_CAT2
- IUMPR_EGR
- IUMPR_EGSENSOR
- IUMPR_EVAP
- IUMPR_FLSYS
- IUMPR_NMHCCAT
- IUMPR_NOXADSORB
- IUMPR_NOXCAT
- IUMPR_OXS1
- IUMPR_OXS2
- IUMPR_PF1
- IUMPR_PF2
- IUMPR_PMFILTER
- IUMPR_PRIVATE
- IUMPR_SAIR
- IUMPR_SECOXS1
- IUMPR_SECOXS2

⌋

[TPS_DEXT_01149] Standardized values of DiagnosticIumprDenominator-Group.category ⌈The semantics of meta-class DiagnosticIumprDenominator-Group is identified by means of the value of attribute DiagnosticIumpr-DenominatorGroup.category.

Standardized values of attribute DiagnosticIumprDenominatorGroup.category are:

- IUMPR_DENOMINATOR_500_MILES
- IUMPR_DENOMINATOR_COLDSTART
- IUMPR_DENOMINATOR_EVAP
- IUMPR_DENOMINATOR_NONE
- IUMPR_DENOMINATOR_PHYSICAL_API
- IUMPR_DENOMINATOR_CSERS
- IUMPR_DENOMINATOR_CSERS_API
- IUMPR_DENOMINATOR_EVAPPURGEFLOW

⌋

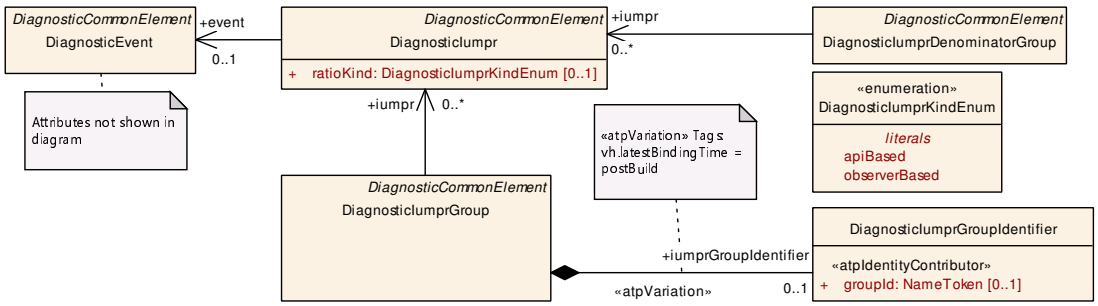

Figure 4.72: Modeling of IUMPR handling for diagnostic events

[TPS_DEXT_01156] Further definition of the semantics of DiagnosticIumpr ⌈The semantics of meta-class DiagnosticIumpr in terms of how the ratio is produced can be further specified by attribute ratioKind.⌋

Class	**Diagnosticlumpr**
Note	This meta-class represents the ability to model the in-use monitor performance ratio. The latter computes to the number of times a fault could have been found divided by the number of times the vehicle conditions have been properly fulfilled. **Tags:** atp.recommendedPackage=Diagnosticlumprs
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*

▽

△

Class	DiagnosticIumpr			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
event	DiagnosticEvent	0..1	ref	This reference represents the DiagnosticEvent that corresponds to the IUMPR computation. **Tags:** atp.Splitkey=event
ratioKind	DiagnosticIumprKind Enum	0..1	attr	This attribute controls the behavior of how the ratio is calculated.

Table 4.208: DiagnosticIumpr

Enumeration	DiagnosticIumprKindEnum
Note	This enumeration is used to control the ratio calculation behavior.
Aggregated by	DiagnosticIumpr.ratioKind
Literal	Description
apiBased	The calculation is based on the usage of an API. **Tags:** atp.EnumerationLiteralIndex=0
observerBased	The calculation is based on the usage of an observer. **Tags:** atp.EnumerationLiteralIndex=1

Table 4.209: DiagnosticIumprKindEnum

Class	DiagnosticIumprGroup			
Note	This meta-class represents the ability to model a IUMPR groups. **Tags:** atp.recommendedPackage=DiagnosticIumprGroups			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
iumpr	DiagnosticIumpr	*	ref	This reference collects DiagnosticIumpr to a DiagnosticIumprGroup. **Tags:** atp.Splitkey=iumpr
iumprGroup Identifier	DiagnosticIumprGroup Identifier	0..1	aggr	This aggregation allows for the variant modeling of the groupIdentifier. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iumprGroupIdentifier, iumprGroup Identifier.groupId, iumprGroupIdentifier.variation Point.shortLabel vh.latestBindingTime=postBuild

Table 4.210: DiagnosticIumprGroup

[constr_1853] Existence of attribute DiagnosticIumprGroup.iumprGroup-Identifier

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticIumprGroup, attribute iumprGroupIdentifier shall exist.⌋

Class	DiagnosticIumprGroupIdentifier			
Note	This meta-class provides the ability to the define the group identifier for an IumprGroup.			
Base	*ARObject*			
Aggregated by	DiagnosticIumprGroup.iumprGroupIdentifier			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
groupId	NameToken	0..1	attr	This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName. **Stereotypes:** atpIdentityContributor

Table 4.211: DiagnosticIumprGroupIdentifier

[constr_1854] Existence of attribute DiagnosticIumprGroupIdentifier.groupId

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticIumprGroupIdentifier, attribute groupId shall exist.⌋

Class	DiagnosticIumprDenominatorGroup			
Note	This meta-class represents the ability to model a IUMPR denominator groups. **Tags:** atp.recommendedPackage=DiagnosticIumprDenominatorGroups			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
iumpr	DiagnosticIumpr	*	ref	This reference collects DiagnosticIumpr to a Diagnostic IumprDenominatorGroup. **Tags:** atp.Splitkey=iumpr

Table 4.212: DiagnosticIumprDenominatorGroup

4.4.14.1 Dem Configuration for OBD-II

[TPS_DEXT_01111] Legislative freeze frame for the OBD-II case ⌈For the implementation of OBD-II, the legislative freeze frame is obtained from running the OBD Mode 0x02 service, modeled by means of meta-class DiagnosticPowertrainFreezeFrame.⌋

Please note that the meta-class DiagnosticTroubleCodeObd is only applicable for the implementation of OBD-II.

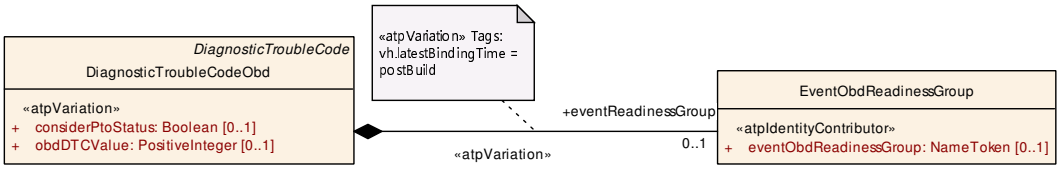

Figure 4.73: Modeling of DTCs for OBD-II

4.4.14.2 Dem Configuration for WWH-OBD

[TPS_DEXT_01112] Definition of a diagnostic trouble code for the implementation of WWH-OBD ⌈The definition of a diagnostic trouble code for the implementation of WWH-OBD implies the existence of a DiagnosticTroubleCodeUds where attribute wwhObdDtcClass is set to any value other than demDtcWwhObdClassNoInformation.⌋

[TPS_DEXT_01093] Definition of legislative freeze frame for WWH-OBD ⌈The legislative WWH-OBD freeze frame is identified by the reference DiagnosticTroubleCodeProps.freezeFrame where attribute recordNumber is set to the value 0.

Optionally, the existence of a reference in the role DiagnosticTroubleCodeProps.extendedDataRecord with attribute recordNumber set to the value 90 is supported.⌋

4.5 Functional Inhibition

4.5.1 Introduction

Conceptually, the FIM (see [13, SWS Function Inhibition Manager]) is closely related to the DEM since it handles the relation of functionality (expressed via the so-called *function identifier*, or in short-form: *Fid*) and linked DiagnosticEvents.

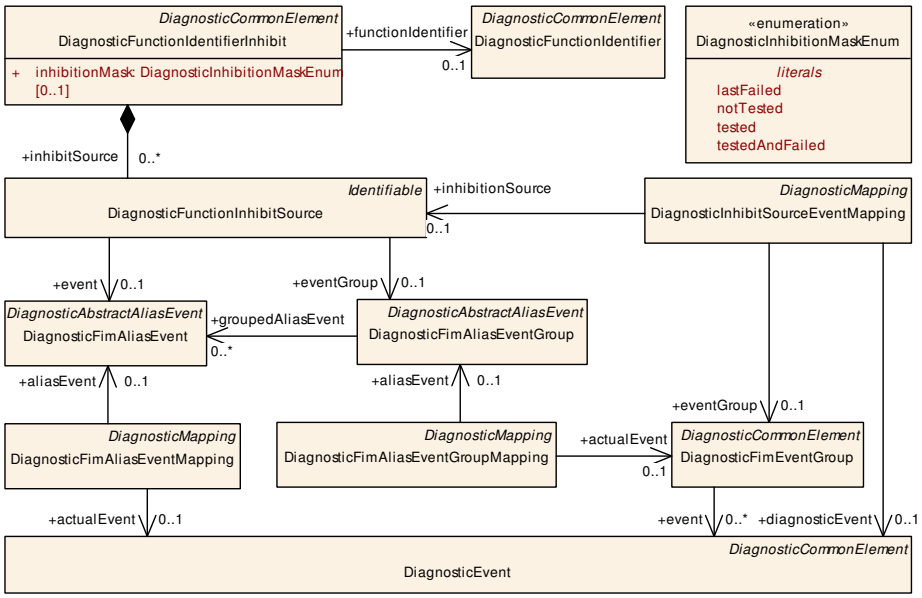

Figure 4.74: Big Picture of FIM diagnostics configuration

An overview of the configuration with regard to function inhibition is depicted in Figure 4.74. The description of the relevant mapping classes related to functional inhibition (e.g. DiagnosticFimAliasEventMapping) is available in Chapter 5.4.

4.5.2 Alias Events

The close relation of FIM and DEM may have consequences in a distributed configuration: it may not be possible to configure the FIM before the DEM is configured because model elements (especially DiagnosticEvent) from the DEM configuration are required to model the FIM configuration.

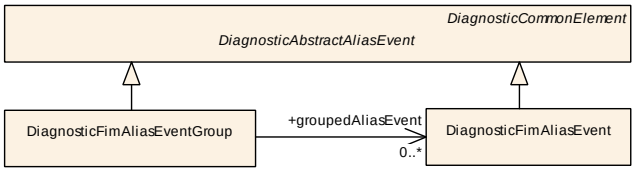

Figure 4.75: Modeling of the DiagnosticAbstractAliasEvent

This leads to the definition of DiagnosticFimAliasEvents that can be taken to model the FIM configuration even if no DEM configuration exists or if configurations with different DEM event names shall be considered for the final projects.

Please note, however, that the definition of DiagnosticFimAliasEvents is not mandatory for the configuration of the FIM. It is possible to directly take the existence of DiagnosticEvents into account and thereby bypass the definition of DiagnosticFimAliasEvents.

Class	**DiagnosticFimAliasEvent**			
Note	This meta-class is used to represent a given event semantics. However, the name of the actual events used in a specific project is sometimes not defined yet, not known or not in the responsibility of the author. Therefore, the DiagnosticFimAliasEvent has a reference to the actual DiagnosticEvent and by this the final connection is created. **Tags:** atp.recommendedPackage=DiagnosticFimAliasEvents			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAbstractAliasEvent*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.213: DiagnosticFimAliasEvent

Class	***DiagnosticAbstractAliasEvent*** (abstract)			
Note	This meta-class represents an abstract base class for all diagnostic alias events.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticFimAliasEvent, DiagnosticFimAliasEventGroup			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.214: DiagnosticAbstractAliasEvent

4.5.3 Function Identifier

The support of the FIM as an extension of the DEM meta-model requires the modeling of additional meta-classes to describe the various ways of possible mappings between DiagnosticEvents and Fids (or, in terms of the meta-model, DiagnosticFunctionIdentifier).

[TPS_DEXT_01121] Semantics of DiagnosticFunctionIdentifier ⌈A DiagnosticFunctionIdentifier can be inhibited by different sources, i.e. in different ways:

Event This corresponds to the DiagnosticEvent.

Fim Event Group This represents a group of events that is defined in the scope of the FIM (within the FIM terminology, this is also known as a *summary event*). In the meta-model, a *Fim Event Group* is represented by means of meta-class DiagnosticFimEventGroup.

To emphasize the locality, the corresponding meta-class has been named DiagnosticFimEventGroup. DiagnosticFimEventGroup is able to reference 0..* DiagnosticEvents in the role event.

This way, the membership of DiagnosticEvents in the hypothetical group of events formed by the DiagnosticFimEventGroup is expressed.

⌋

Please note that DiagnosticEvent is discussed in Chapter 4.4.2. Finally all sources will refer to DiagnosticEvents.

Class	**DiagnosticFunctionIdentifier**			
Note	This meta-class represents a diagnostic function identifier (a.k.a. FID). **Tags:** atp.recommendedPackage=DiagnosticFunctionIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table 4.215: DiagnosticFunctionIdentifier

Class	**DiagnosticFunctionIdentifierInhibit**			
Note	This meta-class represents the ability to define the inhibition of a specific function identifier within the Fim configuration. **Tags:** atp.recommendedPackage=DiagnosticFunctionIdentifierInhibits			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	DiagnosticFunctionIdentifierInhibit			
function Identifier	DiagnosticFunction Identifier	0..1	ref	This represents the corresponding function identifier. **Tags:** atp.Splitkey=functionIdentifier
inhibitionMask	DiagnosticInhibition MaskEnum	0..1	attr	This represents the value of the inhibition mask behavior.
inhibitSource	DiagnosticFunction InhibitSource	*	aggr	This represents a collection of DiagnosticFunctionInhibit Source that contribute to the configuration of the enclosing DiagnosticFunctionIdentiferInhibit. **Tags:** atp.Splitkey=inhibitSource, inhibitSource.short Name

Table 4.216: DiagnosticFunctionIdentifierInhibit

[constr_1855] Existence of the attribute DiagnosticFunctionIdentifier-Inhibit.inhibitionMask

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticFunctionIdentifierInhibit, attribute inhibition-Mask shall exist.⌋

Class	DiagnosticFunctionInhibitSource			
Note	This meta-class represents the ability to define an inhibition source in the context of the Fim configuration.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticFunctionIdentifierInhibit.inhibitSource			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
event	DiagnosticFimAlias Event	0..1	ref	This represents the alias event applicable for the referencing inhibition source. **Tags:** atp.Splitkey=event
eventGroup	DiagnosticFimAlias EventGroup	0..1	ref	This represents the event group applicable for the referencing inhibition source. **Tags:** atp.Splitkey=eventGroup

Table 4.217: DiagnosticFunctionInhibitSource

Enumeration	DiagnosticInhibitionMaskEnum
Note	This meta-class represents the ability to define different kinds of inhibition mask behavior.
Aggregated by	DiagnosticFunctionIdentifierInhibit.inhibitionMask
Literal	**Description**
lastFailed	This represents the inhibition mask behavior "last failed". **Tags:** atp.EnumerationLiteralIndex=0
notTested	This represents the inhibition mask behavior "not tested". **Tags:** atp.EnumerationLiteralIndex=1
tested	This represents the inhibition mask behavior "tested". **Tags:** atp.EnumerationLiteralIndex=3
testedAndFailed	This represents the inhibition mask behavior "tested and failed". **Tags:** atp.EnumerationLiteralIndex=2

Table 4.218: DiagnosticInhibitionMaskEnum

[TPS_DEXT_01096] Semantics of DiagnosticFunctionInhibitSource ⌈The *function inhibition* itself is modeled by means of a different meta-class named DiagnosticFunctionIdentifierInhibit. This meta-class, in turn, references the DiagnosticFunctionIdentifier and it also aggregates 0..* instances of a further meta-class named DiagnosticFunctionInhibitSource.

DiagnosticFunctionInhibitSource inherits from Identifiable in order to be able to utilize attributes like desc, adminData, and introduction for the purpose of textually explaining the reason for defining a certain inhibit source.⌋

[TPS_DEXT_01097] Standardized value of StructuredReq.category for the modeling of DiagnosticFunctionInhibitSource ⌈If DiagnosticFunctionInhibitSource.introduction.structuredReq is used to document the reason for a specific function inhibition, then DiagnosticFunctionInhibitSource.introduction.structuredReq.category shall be set to the value InhibitReason.⌋

[constr_1453] References from DiagnosticFunctionInhibitSource

Imposition time: CP: IT_DiagDes

⌈Each DiagnosticFunctionInhibitSource may either reference one of the following meta-classes in the respective role:

- DiagnosticFimAliasEventMapping in the role event
- DiagnosticFimAliasEventGroupMapping in the role eventGroup

⌋

The semantics of meta-classes DiagnosticFimAliasEventMapping and DiagnosticFimAliasEventGroupMapping is described in Chapter 5.4.2.

Class	**DiagnosticFimEventGroup**			
Note	This meta-class represents the ability to model a Fim event group, also known as a summary event in Fim terminology. This represents a group of single diagnostic events. **Tags:** atp.recommendedPackage=DiagnosticFimEventGroups			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
event	DiagnosticEvent	*	ref	This reference represents the way of grouping diagnostic events into a summary event in the context of the Fim. **Tags:** atp.Splitkey=event

Table 4.219: DiagnosticFimEventGroup

[TPS_DEXT_01098] Semantics of attribute DiagnosticFunctionInhibitSource.event ⌈If the reference DiagnosticFunctionInhibitSource.event exists this means the inhibition applies to a single DiagnosticEvent.⌋

[TPS_DEXT_01099] **Semantics of attribute `DiagnosticFunctionInhibitSource.eventGroup`** ⌈If the reference `DiagnosticFunctionInhibitSource.eventGroup` exists this means that the inhibition effectively applies to a group of `DiagnosticEvent`s where the actual grouping is defined in the scope of the `FIM`.⌋

4.6 Diagnostics on J1939

4.6.1 Introduction

Diagnostics on J1939 is to some extent similar to the “rest of the world”. In other words, J1939 uses a Dcm and a Dem similarly as other communication networks do.

On the other hand, there are significant differences between J1939 and the “rest of the world” that justify the creation of a separate chapter that focuses entirely on how a diagnostic extract for J1939 shall look like.

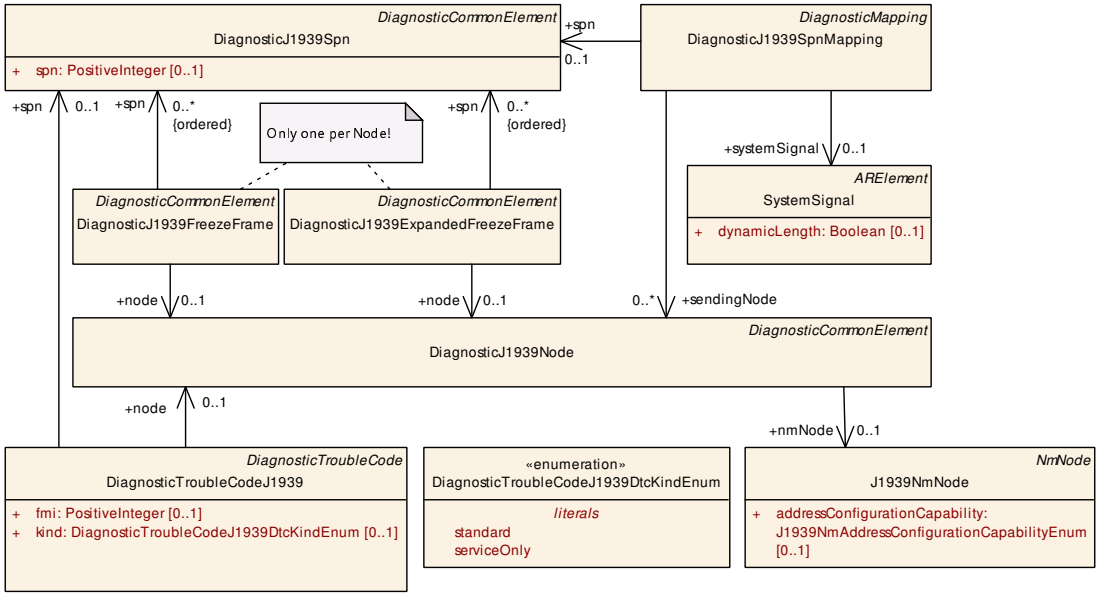

Figure 4.76: Big Picture of J1939 diagnostics configuration

That said, a support for Dcm-related model elements for J1939 is mainly happening in the domain of the [5, TPS System Template] rather than in the scope of this document.

The actual configuration of services supported by the J1939Dcm is done by assigning DMx messages in the System Description.

The mapping of the `J1939DcmIPdu`s to the diagnostic services of J1939 Dcm happens during derivation to `EcuC`.

In the following sections this document focuses on explaining the modeling with respect to the Dem.

An overview to the configuration of J1939 diagnostics is depicted in Figure 4.76. The description of the relevant mapping classes related to J1939 (e.g. DiagnosticJ1939SpnMapping) is available in Chapter 5.5.

4.6.2 Suspect Parameter Number

In the J1939 terminology, a *Suspect Parameter Number* represents a signal identifier. The numerical value of this signal identifier is stored in the attribute DiagnosticJ1939Spn.spn.

Conceptually, the *Suspect Parameter Number* binds to the SystemSignal, i.e. by attaching a *Suspect Parameter Number* to a SystemSignal a J1939 signal is created.

At first sight, it may seem like a good idea to attach the spn attribute to the SystemSignal itself. However, this would place a very specific J1939-related model semantics in a very prominent place.

This just doesn't seem right and thanks to the existence of meta-class DiagnosticJ1939SpnMapping (see Chapter 5.5.1) it is possible to "inject" the J1939 signal identifier into a SystemSignal without actually touching the SystemSignal.

Class	**DiagnosticJ1939Spn**			
Note	This meta-class represents the ability to model a J1939 Suspect Parameter Number (SPN). **Tags:** atp.recommendedPackage=DiagnosticJ1939Spns This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
spn	PositiveInteger	0..1	attr	This attribute represents the concrete numerical identification for the enclosing SPN.

Table 4.220: DiagnosticJ1939Spn

[constr_1856] Existence of attribute DiagnosticJ1939Spn.spn

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticJ1939Spn, attribute spn shall exist.⌋

4.6.3 Dem-related Modeling

The modeling of Dem-related support for J1939 is centered around the DiagnosticJ1939Node. This meta-class literally acts as a sort of "inverted"[8] hub for all the Dem-related model elements. The DiagnosticJ1939Node represents a specific function, or in terms of the J1939 terminology, a *Controller Application*.

This aspect is stressed in Figure 4.76.

[8]Meaning: several meta-classes are referencing the DiagnosticJ1939Node.

Since, according to the J1939 concept, each *controller application* represents an independent entity in terms of network management, the following constraint applies:

[constr_1455] Relation of DiagnosticJ1939Node to J1939NmNode

Imposition time: CP: IT_DiagDes

⌈Each J1939NmNode shall only be referenced in the role nmNode by a single DiagnosticJ1939Node.⌋

J1939 supports the capturing of information similar to the definition of DiagnosticFreezeFrame, as explained in Chapter 4.4.6. In the case of J1939, however, two different kinds of freeze frames can be defined:

[TPS_DEXT_01104] Difference between DiagnosticJ1939FreezeFrame and DiagnosticJ1939ExpandedFreezeFrame ⌈A DiagnosticJ1939FreezeFrame is transmitted via J1939 **DM04** while a DiagnosticJ1939ExpandedFreezeFrame is transmitted via J1939 **DM24**/**DM25**.⌋

[TPS_DEXT_01105] Relation of DiagnosticJ1939Spn to DiagnosticJ1939FreezeFrame and DiagnosticJ1939ExpandedFreezeFrame ⌈It is possible that a given DiagnosticJ1939Spn is referenced by both a DiagnosticJ1939FreezeFrame and a DiagnosticJ1939ExpandedFreezeFrame.

In other words, the *Suspect Parameter Number* can be part of a normal freeze frame and, at the same time, an expanded freeze frame.⌋

Class	**DiagnosticJ1939FreezeFrame**			
Note	This meta-class represents the ability to model a J1939 Freeze Frame. **Tags:** atp.recommendedPackage=DiagnosticJ1939FreezeFrames This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
node	DiagnosticJ1939Node	0..1	ref	This represents the DiagnosticJ1939Node to which the J1939 freeze frame is associated. **Tags:** atp.Splitkey=node
spn (ordered)	DiagnosticJ1939Spn	*	ref	This represents the collection of SPNs that make the J1939 Freeze Frame.

Table 4.221: DiagnosticJ1939FreezeFrame

Class	**DiagnosticJ1939ExpandedFreezeFrame**
Note	This meta-class represents the ability to model an expanded J1939 Freeze Frame. **Tags:** atp.recommendedPackage=DiagnosticJ1939ExpandedFreezeFrames This Class is only used by the AUTOSAR Classic Platform.
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*

▽

△

Class	**DiagnosticJ1939ExpandedFreezeFrame**			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
node	DiagnosticJ1939Node	0..1	ref	This represents the DiagnosticJ1939Node to which the J1939 expanded freeze frame is associated. **Tags:** atp.Splitkey=node
spn (ordered)	DiagnosticJ1939Spn	*	ref	This represents the collection of SPNs that make the expanded J1939 Freeze Frame.

Table 4.222: DiagnosticJ1939ExpandedFreezeFrame

Enumeration	**DiagnosticTroubleCodeJ1939DtcKindEnum**
Note	This meta-class represents the ability to further specify a J1939 DTC in terms of its semantics. This Enumeration is only used by the AUTOSAR Classic Platform.
Aggregated by	DiagnosticTroubleCodeJ1939.kind
Literal	*Description*
serviceOnly	this represents a DTC that is only relevant for service in a garage, reported by e.g. DM53. **Tags:** atp.EnumerationLiteralIndex=0
standard	This represents a non-specific DTC reported by e.g. DM1. **Tags:** atp.EnumerationLiteralIndex=1

Table 4.223: DiagnosticTroubleCodeJ1939DtcKindEnum

[constr_1456] Valid interval for attribute `DiagnosticTroubleCodeJ1939.fmi`

Imposition time: CP: `IT_DiagDes`

⌈The value of the attribute `DiagnosticTroubleCodeJ1939.fmi` shall be in the interval 0..31.⌋

Please note that the rationale for the existence of [constr_1456] as well as the meaning of the individual values of the attribute `fmi` can be found in the respective [21, SAE J1939] specification.

Class	**DiagnosticTroubleCodeJ1939**			
Note	This meta-class represents the ability to model specific trouble-code related properties for J1939. **Tags:** atp.recommendedPackage=DiagnosticTroubleCodes This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticTroubleCode*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
dtcProps	DiagnosticTroubleCodeProps	0..1	ref	Defined properties associated with the J1939 DTC. **Tags:** atp.Splitkey=dtcProps
fmi	PositiveInteger	0..1	attr	This attribute represents the behavior of the Failure Mode Indicator.
kind	DiagnosticTroubleCodeJ1939DtcKindEnum	0..1	attr	This attribute further specifies the DTC in terms of its semantics.

▽

△

Class	**DiagnosticTroubleCodeJ1939**			
node	DiagnosticJ1939Node	0..1	ref	This represents the related DiagnosticJ1939Node. **Tags:** atp.Splitkey=node
spn	DiagnosticJ1939Spn	0..1	ref	This represents the releated SPN. **Tags:** atp.Splitkey=spn

Table 4.224: DiagnosticTroubleCodeJ1939

[constr_1780] Existence of attribute `DiagnosticTroubleCodeJ1939.fmi`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticTroubleCodeJ1939`, attribute `fmi` shall exist.⌋

[constr_1781] Existence of attribute `DiagnosticTroubleCodeJ1939.spn`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticTroubleCodeJ1939`, attribute `spn` shall exist.⌋

[TPS_DEXT_01107] Definition of service-only DTC ⌈It is possible to specify whether given `DTC`s on J1939 are only relevant for a service session in a garage or workshop. The common property of service-only `DTC`s is that they shall be located in a memory section that is exclusively used by service-only `DTC`s.⌋

The statement made by [TPS_DEXT_01107] needs to be secured by two constraints:

[constr_1457] Service-only DTCs shall refer to a common memory section

Imposition time: CP: `IT_DiagDes`

⌈All `DiagnosticTroubleCodeJ1939` with attribute `kind` set to the value `serviceOnly` that reference the same `DiagnosticJ1939Node` shall also reference the same `DiagnosticTroubleCodeProps.diagnosticMemory`.⌋

[constr_1458] Reference to `DiagnosticMemoryDestination`

Imposition time: CP: `IT_DiagDes`

⌈A `DiagnosticMemoryDestination` that is referenced by a `DiagnosticTroubleCodeJ1939.dtcProps.diagnosticMemory` where the value of attribute `DiagnosticTroubleCodeJ1939.kind` is set to `serviceOnly` shall **not be referenced by any other** `DiagnosticTroubleCodeJ1939` where attribute `kind` is set to any other value than `serviceOnly`.⌋

5 Diagnostic Mapping

5.1 Introduction

This chapter contains a description of how specific diagnostic resources can be associated

- with each other to form a new kind of model semantics that goes beyond the meaning the individual diagnostic resources that are affected by the combination, typically implemented by means of a specialized subclass of `DiagnosticMapping`, or
- with model elements in either the application software or other places in the basic software stack. In this case the connection is typically created by means of subclasses of `DiagnosticSwMapping`.

The mapping concept of the `DiagnosticExtract` template has been designed to support the decentralized and independent definition of diagnostic requirements that can be linked together at a late point during the development process.

It also supports the use of mapping contributions collected from various sources in order to reduce manual mapping work by the `ECU` integrator.

Class	***DiagnosticMapping*** (abstract)			
Note	Abstract element for different kinds of diagnostic mappings.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	CpSwClusterResourceToDiagDataElemMapping, CpSwClusterResourceToDiagFunctionIdMapping, CpSwClusterToDiagEventMapping, CpSwClusterToDiagRoutineSubfunctionMapping, DiagnosticAuthTransmitCertificateMapping, DiagnosticEdrDataProviderMapping, DiagnosticEventToDebounceAlgorithmMapping, DiagnosticEventToEnableConditionGroupMapping, DiagnosticEventToOperationCycleMapping, DiagnosticEventToSecurityEventMapping, DiagnosticEventToStorageConditionGroupMapping, DiagnosticEventToTroubleCodeJ1939Mapping, DiagnosticEventToTroubleCodeUdsMapping, DiagnosticFimAliasEventGroupMapping, DiagnosticFimAliasEventMapping, DiagnosticInhibitSourceEventMapping, DiagnosticIumprToFunctionIdentifierMapping, DiagnosticJ1939SpnMapping, DiagnosticMasterToSlaveEventMapping, DiagnosticSecureCodingMapping, *DiagnosticSwMapping*, DiagnosticTroubleCodeUdsToTroubleCodeObdMapping			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
provider SoftwareCluster	CpSoftwareCluster	0..1	ref	This reference can be used in an early design phase to associate an element of a diagnostic extract with an existing provided CPSoftwareCluster. **Tags:** atp.Splitkey=providerSoftwareCluster atp.Status=draft This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	*DiagnosticMapping* (abstract)			
requester SoftwareCluster	CpSoftwareCluster	0..1	ref	This reference can be used in an early design phase to associate an element of a diagnostic extract with an existing requested CPSoftwareCluster. **Tags:** atp.Splitkey=requesterSoftwareCluster atp.Status=draft This Attribute is only used by the AUTOSAR Classic Platform.

Table 5.1: DiagnosticMapping

The description of the modeling of diagnostic resources can be roughly divided into five sections:

- A section dedicated to the mapping of diagnostic services, see Chapter 5.2.
- A section dedicated to the mapping in the context of event handling and fault memory handling, see Chapter 5.3.
- A section dedicated to the mapping in the context of the functional inhibition, see Chapter 5.4.
- A section dedicated to the mapping in the context of the diagnostics on J1939, see Chapter 5.5.
- A section dedicated to the mapping in the context of the software-clusters on the *AUTOSAR classic Platform*, see Chapter 5.6.

5.2 Mapping of Diagnostic Services

Automotive diagnostics interacts with both application software and basic software in various ways that can be formalized using the AUTOSAR meta-model. This chapter contains a description of the formalization of this interaction along with the applicable constraints.

[TPS_DEXT_01040] Use case where the `DiagnosticExtract` refers to software-components ⌈This is a list of the potential use cases where the `DiagnosticExtract` refers to software-components in general and `PortPrototype`s in the context of either `CompositionSwComponentType`s or `AtomicSwComponentType`s:

- `DiagnosticExtract` refers to `PortPrototype` (for the access to `dataElement`) or `SwcServiceDependency` in the context of a `AtomicSwComponentType` embedded in the hierarchy created by the `rootSoftwareComposition`.
- `DiagnosticExtract` refers to a `PortPrototype` (for the access to `dataElement`) or `SwcServiceDependency` in the context of a `AtomicSwComponentType` embedded in the hierarchy created by a `CompositionSwComponentType` that is nowhere aggregated (for the time being).

- DiagnosticExtract refers to a PortPrototype (for the access to dataElement) or SwcServiceDependency in the context of an AtomicSwComponentType.
- DiagnosticExtract refers to a BswServiceDependency.

⌋

Class	SwcServiceDependency			
Note	Specialization of ServiceDependency in the context of an SwcInternalBehavior. It allows to associate ports, port groups and (in special cases) data defined for an atomic software component to a given ServiceNeeds element.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceDependency*			
Aggregated by	AdaptiveSwcInternalBehavior.serviceDependency, *AtpClassifier*.atpFeature, SwcInternal Behavior.serviceDependency			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
assignedData	RoleBasedData Assignment	*	aggr	Defines the role of an associated data object of the same component. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=assignedData, assignedData.variation Point.shortLabel vh.latestBindingTime=preCompileTime
assignedPort	RoleBasedPort Assignment	*	aggr	Defines the role of an associated port of the same component. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=assignedPort, assignedPort.variation Point.shortLabel vh.latestBindingTime=preCompileTime
representedPort Group	PortGroup	0..1	ref	This reference specifies an association between the ServiceNeeeds and a PortGroup, for example to request a communication mode which applies for communication via these ports. The referred PortGroup shall be local to this atomic SWC, but via the links between the Port Groups, a tool can evaluate this information such that all the ports linked via this port group on the same ECU can be found. **Tags:** atp.Splitkey=representedPortGroup
serviceNeeds	ServiceNeeds	0..1	aggr	The associated ServiceNeeds. **Tags:** atp.Splitkey=serviceNeeds, serviceNeeds.short Name

Table 5.2: SwcServiceDependency

Class	BswServiceDependency			
Note	Specialization of ServiceDependency in the context of an BswInternalBehavior. It allows to associate BswModuleEntries and data defined for a BSW module or cluster to a given ServiceNeeds element.			
Base	*ARObject*, *ServiceDependency*			
Aggregated by	BswInternalBehavior.serviceDependency			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**BswServiceDependency**			
assignedData	RoleBasedData Assignment	*	aggr	Defines the role of an associated data object (owned by this module or cluster) in the context of the ServiceNeeds element. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=assignedData, assignedData.variation Point.shortLabel vh.latestBindingTime=preCompileTime
assignedEntry Role	RoleBasedBswModule EntryAssignment	*	aggr	Defines the role of an associated BswModuleEntry in the context of the ServiceNeeds element. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=assignedEntryRole, assignedEntry Role.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
ident	BswService DependencyIdent	0..1	aggr	This adds the ability to become referrable to BswService Dependency. **Stereotypes:** atpIdentityContributor **Tags:** xml.sequenceOffset=-100
serviceNeeds	ServiceNeeds	0..1	aggr	The associated ServiceNeeds. **Tags:** atp.Splitkey=serviceNeeds, serviceNeeds.short Name

Table 5.3: BswServiceDependency

Figure 5.1 provides an overview of mapping classes related to diagnostic services.

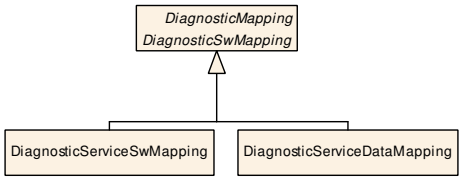

Figure 5.1: Modeling of the `DiagnosticMapping` in the context of diagnostic services

The references in the roles

- `DiagnosticServiceDataMapping.diagnosticParameter`
- `DiagnosticServiceDataMapping.diagnosticDataElement`
- `DiagnosticServiceSwMapping.diagnosticParameter`
- `DiagnosticServiceSwMapping.diagnosticDataElement`

would technically have to be made variant because of the variant aggregation

- `DiagnosticDataIdentifier.dataElement` resp.
- `DiagnosticParameterIdentifier.dataElement` and finally
- `DiagnosticAbstractParameter.dataElement`.

However, such a change cannot be achieved in a backwards-compatible way and the less elegant, but existing mechanism of implementing the variation points (where applicable) by means of referencing

- DiagnosticServiceDataMapping and
- DiagnosticServiceSwMapping

from DiagnosticContributionSet in the role element is considered less harmful in this specific case.

In other words, the missing variability in the mentioned references shall be compensated by creating additional DiagnosticServiceDataMapping and DiagnosticServiceSwMapping and creating references to DiagnosticDataElement resp. DiagnosticParameterIdent from these new instances of the mapping classes.

5.2.1 Diagnostic Service Data Mapping

Please note that the Dcm is in general entitled to both read and write a dataElement. This applies even if the corresponding PortPrototype is a PPortPrototype. This means that the diagnostic service data mapping is limited to SenderReceiverInterface.

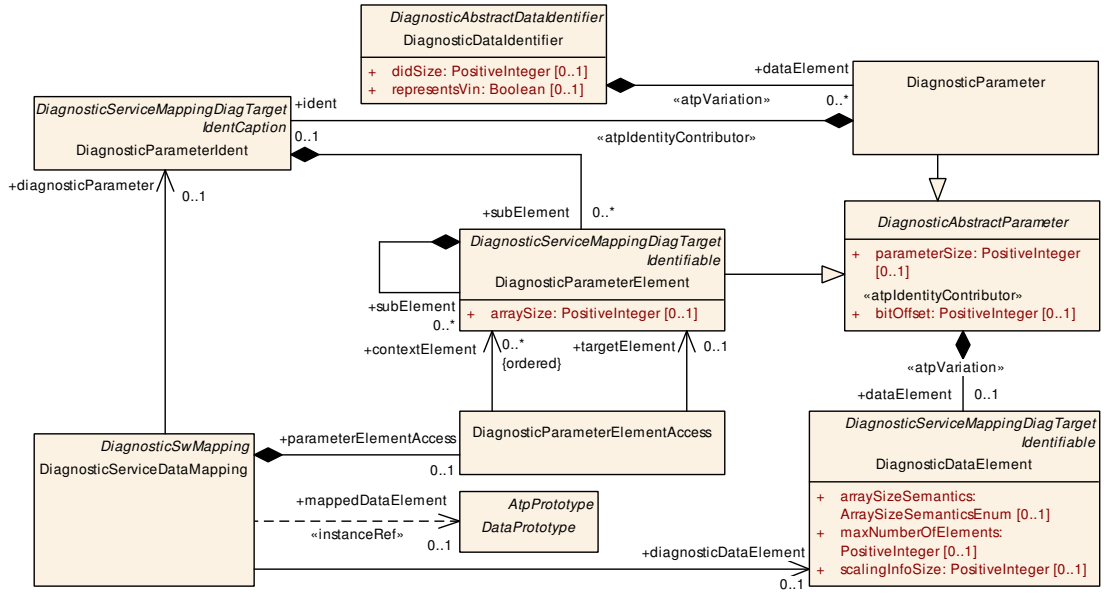

Figure 5.2: Dcm service data mapping

[TPS_DEXT_01178] Semantics of DiagnosticServiceDataMapping.parameterElementAccess ⌈A reference to a given DiagnosticParameterElement may need more than one reference in order to be able to resolve the element position in nested array-shaped DiagnosticParameterElements.

Therefore the meta-class DiagnosticParameterElementAccess utilizes the following references:

- contextElement covers the ability to provide references to array elements in nested array-shaped DiagnosticParameterElements

- targetElement defines the reference to the final target DiagnosticParameterElement.

⌋

For clarification, this means that a reference to a DiagnosticParameterElement inside a deeply nested structure of DiagnosticParameterElements **only requires the targetElement if no array-semantics exists** on any intermediary level of the nested DiagnosticParameterElements.

The reference in the role contextElement is only required to identify array elements on intermediary levels of the nested structure.

[TPS_DEXT_01041] Semantics of attribute DiagnosticServiceDataMapping.diagnosticDataElement ⌈By means of the attribute DiagnosticServiceDataMapping.diagnosticDataElement it is possible to specify that the Dcm has access to

- a dataElement in a PortPrototype typed by a SenderReceiverInterface or
- an nvData in a PortPrototype typed by an NvDataInterface.

This type of data access is suitable for the diagnostic services ReadDataByIdentifier (0x22), WriteDataByIdentifier (0x2E), and InputOutputControl (0x2F).⌋

Class	**DiagnosticServiceDataMapping**			
Note	This represents the ability to define a mapping of a diagnostic service to a software-component. This kind of service mapping is applicable for the usage of SenderReceiverInterfaces or event/notifier semantics in ServiceInterfaces on the adaptive platform. **Tags:** atp.recommendedPackage=DiagnosticServiceMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnosticData Element	DiagnosticDataElement	0..1	ref	This represents the applicable payload that corresponds to the referenced DataPrototype in the role mappedData Element or (in case of a usage on the adaptive platform) mappedApDataElement. **Tags:** atp.Splitkey=diagnosticDataElement
diagnostic Parameter	DiagnosticParameter Ident	0..1	ref	This represents the applicable payload that corresponds to the referenced DataPrototype in the role mappedData Element. **Tags:** atp.Splitkey=diagnosticParameter xml.sequenceOffset=20

▽

△

Class	**DiagnosticServiceDataMapping**			
mappedData Element	DataPrototype	0..1	iref	This represents the dataElement in the application software that is accessed for diagnostic purpose. This role is applicable on the classic platform. **Tags:** atp.Splitkey=mappedDataElement, mappedData Element.contextRootComposition, mappedData Element.contextComponent, mappedData Element.contextPort, mappedDataElement.rootData Prototype, mappedDataElement.contextDataPrototype, mappedDataElement.targetDataPrototype **InstanceRef implemented by:** DataPrototypeInSystem InstanceRef This Attribute is only used by the AUTOSAR Classic Platform.
parameter ElementAccess	DiagnosticParameter ElementAccess	0..1	aggr	This aggregation represents the single point of access to the reference to one specific DiagnosticParameter Element. **Tags:** atp.Splitkey=parameterElementAccess

Table 5.4: DiagnosticServiceDataMapping

Class	**DiagnosticParameterElementAccess**			
Note	This meta-class acts as a single point for defining structured references to a specific Diagnostic ParameterElement.			
Base	*ARObject*			
Aggregated by	DiagnosticServiceDataMapping.parameterElementAccess, DiagnosticServiceSwMapping.parameter ElementAccess			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
contextElement (ordered)	DiagnosticParameter Element	*	ref	This represents the context of an applicable payload that corresponds to the referenced DataPrototype in the role mappedDataElement. **Tags:** xml.sequenceOffset=10
targetElement	DiagnosticParameter Element	0..1	ref	This represents the target reference of an applicable payload that corresponds to the referenced Data Prototype in the role mappedDataElement. **Tags:** atp.Splitkey=targetElement xml.sequenceOffset=20

Table 5.5: DiagnosticParameterElementAccess

[constr_1828] Existence of references from `DiagnosticServiceDataMapping`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticServiceDataMapping`, the following references shall exist:

- Reference to a `DiagnosticServiceMappingDiagTarget`, i.e. one of
  - Reference to `DiagnosticDataElement` in the role `diagnosticDataElement`
  - Reference to `DiagnosticParameterIdent` in the role `diagnosticParameter`
  - Reference to `DiagnosticParameterElement` in the roles

  * contextElement (optional)
  * targetElement

  from within parameterElementAccess

- Reference to DataPrototype in the role mappedDataElement

⌋

[constr_1343] Simultaneous existence of the attributes DiagnosticServiceDataMapping.diagnosticDataElement and DiagnosticDataByIdentifier.dataIdentifier

Imposition time: CP: IT_DiagDes

⌈A DiagnosticServiceDataMapping.diagnosticDataElement shall also be aggregated by a DiagnosticDataByIdentifier in the role dataIdentifier.dataElement.dataElement.⌋

[constr_1343] shall only apply for the step in the methodology where the DiagnosticExtract is considered complete enough for the configuration of the Dcm and Dem to be derived. Any intermediate step, e.g. hand-over from OEM to tier-1 supplier does not actually enforce [constr_1343].

In other words, [constr_1343] ensures that there is a connection between the DiagnosticServiceDataMapping and the corresponding DiagnosticReadDataByIdentifier or DiagnosticWriteDataByIdentifier.

Only then does the diagnostic service become fully usable.

The modeling of the reference DiagnosticServiceDataMapping.mappedDataElement only supports the case where the target DataPrototype is typed by an ApplicationDataType.

Please refer to [8, TPS Software Component Template] for a detailed explanation of the meaning of the value of a data type category.

[constr_1345] DiagnosticDataElement shall not (finally) be aggregated by a DiagnosticRoutine

Imposition time: CP: IT_DiagDes

⌈A DiagnosticDataElement that is referenced by a DiagnosticServiceDataMapping shall not (finally) be aggregated by a DiagnosticRoutine.⌋

Class	**ApplicationPrimitiveDataType**
Note	A primitive data type defines a set of allowed values. **Tags:** atp.recommendedPackage=ApplicationDataTypes
Base	*ARElement*, *ARObject*, *ApplicationDataType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*

▽

△

Class	ApplicationPrimitiveDataType			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 5.6: ApplicationPrimitiveDataType

[TPS_DEXT_01042] Dem uses `DiagnosticServiceDataMapping` ⌈There is a use case for the Dem to utilize a `DiagnosticServiceDataMapping` such that elements of a `DiagnosticExtendedDataRecord` are fetched from `dataElement`s in an `ApplicationSwComponentType`.

Therefore, [constr_1345] intentionally does not exclude the aggregation of `DiagnosticDataElement` by `DiagnosticExtendedDataRecord` in the context of `DiagnosticServiceDataMapping`.⌋

Class	ApplicationSwComponentType			
Note	The `ApplicationSwComponentType` is used to represent the application software. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, *ARObject*, *AtomicSwComponentType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *Collectable Element*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 5.7: ApplicationSwComponentType

[constr_1584] `DiagnosticDataElement` shall not be used more than once in I/O Control instance

Imposition time: CP: `IT_DiagDes`

⌈A given `DiagnosticDataElement` shall not be used by more than one `DiagnosticServiceDataMapping` that in turn refers to a `DataPrototype` defined in the context of a `DataInterface` that is used to type a `PortPrototype` that in turn is referenced by a `RoleBasedPortAssignment` where attribute `role` is set to the value `IOControlRequest`.⌋

[TPS_DEXT_01179] Existence of reference to `DiagnosticParameterIdent` in the role `diagnosticParameter` ⌈If the reference `DiagnosticServiceDataMapping.diagnosticParameter` exists, then the intention is to map a composite `DataPrototype` to a `DiagnosticParameter` in one piece.

This means that the elements of the composite `DataPrototype` would be placed one after the other starting at the `bitOffset` of the referenced `DiagnosticParameter`.⌋

Please find examples for the application of `DiagnosticServiceDataMapping` to the nested definition of `DiagnosticParameter`s in Chapter B.2.

5.2.2 Diagnostic Service Data Mapping for OBD

[constr_1450] Service mapping for OBD mode 0x01 for `DiagnosticParameterIdentifier`

Imposition time: CP: `IT_DiagDes`

⌈If a `DiagnosticServiceSwMapping` or `DiagnosticServiceDataMapping` refers to a `DiagnosticRequestCurrentPowertrainData` and a `DiagnosticDataElement` that is aggregated by a `DiagnosticParameterIdentifier` then one of two alternative model configurations shall exist:

- `SwcServiceDependency` referenced by the same `DiagnosticServiceSwMapping` or `DiagnosticServiceDataMapping` shall aggregate an `ObdPidServiceNeeds` in the role `serviceNeeds`.
- The `BswServiceDependencyIdent` referenced by the same `DiagnosticServiceSwMapping` shall aggregate an `ObdPidServiceNeeds` in the role `serviceNeeds`.

⌋

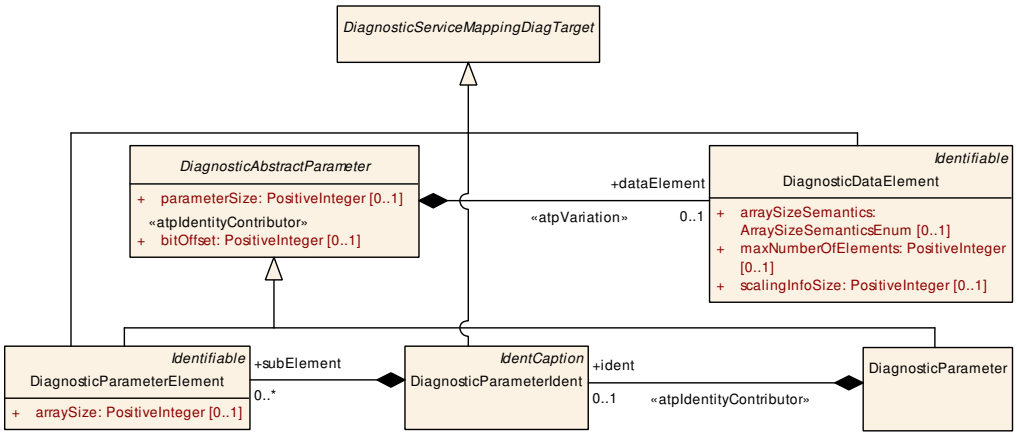

Figure 5.3: Abstract Base class `DiagnosticServiceMappingDiagTarget` for reference targets on the diagnostic end

[constr_1451] Service mapping for OBD mode 0x09 for `DiagnosticInfoType`

Imposition time: CP: `IT_DiagDes`

⌈If a `DiagnosticServiceSwMapping` refers to a `DiagnosticRequestVehicleInfo` and a `DiagnosticDataElement` that is aggregated by a `DiagnosticInfoType` then one of two alternative model configurations shall exist:

- The `SwcServiceDependency` referenced by the same `DiagnosticServiceSwMapping` shall aggregate a `ObdInfoServiceNeeds` in the role `serviceNeeds`.
- The `BswServiceDependencyIdent` referenced by the same `DiagnosticServiceSwMapping` shall aggregate an `ObdInfoServiceNeeds` in the role `serviceNeeds`.

⌋

[constr_1452] Service mapping for OBD mode 0x08 for `DiagnosticInfoType`

Imposition time: CP: `IT_DiagDes`

⌈If a `DiagnosticServiceSwMapping` refers to a `DiagnosticRequestControlOfOnBoardDevice` then the `SwcServiceDependency` referenced by the same `DiagnosticServiceSwMapping` shall aggregate an `ObdControlServiceNeeds` in the role `serviceNeeds`.⌋

Class	**ObdInfoServiceNeeds**			
Note	Specifies the abstract needs of a component or module on the configuration of OBD Services in relation to a given InfoType (OBD Service 09) which is supported by this component or module.			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 5.8: ObdInfoServiceNeeds

Class	**ObdPidServiceNeeds**			
Note	Specifies the abstract needs of a component or module on the configuration of OBD Services in relation to a particular PID (parameter identifier) which is supported by this component or module. In case of using a client/server communicated value, the related value shall be communicated via the port referenced by assignedPort. The details of this communication (e.g. appropriate naming conventions) are specified in the related software specifications (SWS).			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 5.9: ObdPidServiceNeeds

Class	**ObdControlServiceNeeds**			
Note	Specifies the abstract needs of a component or module on the configuration of OBD Service 08 (request control of on-board system) in relation to a particular test-Identifier (TID) supported by this component or module.			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 5.10: ObdControlServiceNeeds

[TPS_DEXT_01177] Utility of `DiagnosticServiceMappingDiagTarget` ⌈The mapping between data in the application software and elements of a `DiagnosticAbstractDataIdentifier`, as implemented by `DiagnosticServiceSwMapping` and `DiagnosticServiceDataMapping`, supports different access points on the diagnostic end.

To facilitate the handling and description of the relationship, abstract class DiagnosticServiceMappingDiagTarget has been defined.⌋

The modeling of the DiagnosticServiceMappingDiagTarget is sketched in Figure 5.3.

Leveraging the concept of the DiagnosticServiceMappingDiagTarget, different cases have to be considered in the configuration of this mapping, depending on the direction of the communication.

Class	***DiagnosticServiceMappingDiagTarget*** (abstract)			
Note	This meta-class serves as a base class for diagnostics-related targets of subclasses of DiagnosticSw Mapping			
Base	*ARObject*			
Subclasses	DiagnosticDataElement, DiagnosticParameterElement, DiagnosticParameterIdent			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 5.11: DiagnosticServiceMappingDiagTarget

The first case affects the reading of data from the application into a DID. The applicable modeling options for the creation of the DiagnosticServiceSwMapping and DiagnosticServiceDataMapping in this case are described by [TPS_DEXT_01184].

In [TPS_DEXT_01184], the column labeled “Symmetric Read/Write possible” indicates whether it would be possible to write the received information back **to** the application.

[TPS_DEXT_01184] Use cases for reading data from the application ⌈

Source **mappedDataElement/accessed-DataPrototype**	**Target** **DiagnosticServiceMappingDiagTarget**	**Symmetric Read/Write possible**
primitive or flat array	DiagnosticDataElement	Yes
composite	DiagnosticParameterIdent	Yes
element of composite	DiagnosticDataElement	No
element of composite	DiagnosticParameterElement	No

⌋

The second case affects the writing of data from a DID to the application. The applicable modeling options for the creation of the DiagnosticServiceSwMapping and DiagnosticServiceDataMapping in this case are described by [TPS_DEXT_01185].

In [TPS_DEXT_01185], the column labeled “Symmetric Read/Write possible” indicates whether it would be possible to read the received information back **from** the application.

[TPS_DEXT_01185] Use cases for writing data to the application ⌈

Source DiagnosticServiceMappingDiagTarget	Target mappedDataElement/accessed-DataPrototype	Symmetric Read/Write possible
DiagnosticDataElement	primitive or flat array	Yes
DiagnosticParameterIdent	composite	Yes
DiagnosticParameterElement	Element of composite	No
DiagnosticParameterElement	primitive	No

⌋

5.2.3 Diagnostic Service Software Mapping

The diagnostic service software mapping is limited to ClientServerInterface or a direct function call (in the case of basic software or complex driver).

Class	**ClientServerInterface**			
Note	A client/server interface declares a number of operations that can be invoked on a server by a client. **Tags:** atp.recommendedPackage=PortInterfaces			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
operation	ClientServerOperation	*	aggr	ClientServerOperation(s) of this ClientServerInterface. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=operation, operation.shortName, operation.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime This Attribute is only used by the AUTOSAR Classic Platform.
possibleError	ApplicationError	*	aggr	Application errors that are defined as part of this interface. **Tags:** atp.Splitkey=possibleError, possibleError.shortName

Table 5.12: ClientServerInterface

[TPS_DEXT_01043] Purpose of DiagnosticServiceSwMapping ⌈The meta-class DiagnosticServiceSwMapping has been introduced to support the creation of a relationship between the definition of a given diagnostic service and the SwcService-Dependency (if the service applies to the application software) or BswService-Dependency (if the service applies to the basic software).⌋

It is required to use the applicable form of reference to the target SwcService-Dependency depending on the context of the enclosing AtomicSwComponentType.

Access to the data in the application is implemented using client/server communication, i.e. the information is passed by means of ArgumentDataPrototypes owned by ClientServerOperations.

In principle, the information obtained from the application can be typed by a composite data type of arbitrary structure. The composite data type defined in the application context may be associated with a nested structure inside the corresponding DiagnosticAbstractDataIdentifier (resp. DiagnosticRoutineSubfunction) and it may happen that only a subset of the composite data type is associated with the DiagnosticParameterElements of the nested definition of the DiagnosticAbstractDataIdentifier/DiagnosticRoutineSubfunction.

Therefore, in the case of a mapping from complex type to nested DID structure, the purpose of the DiagnosticServiceSwMapping is not only to identify the DiagnosticParameterElement, but also the DataPrototype inside a composite ArgumentDataPrototype.

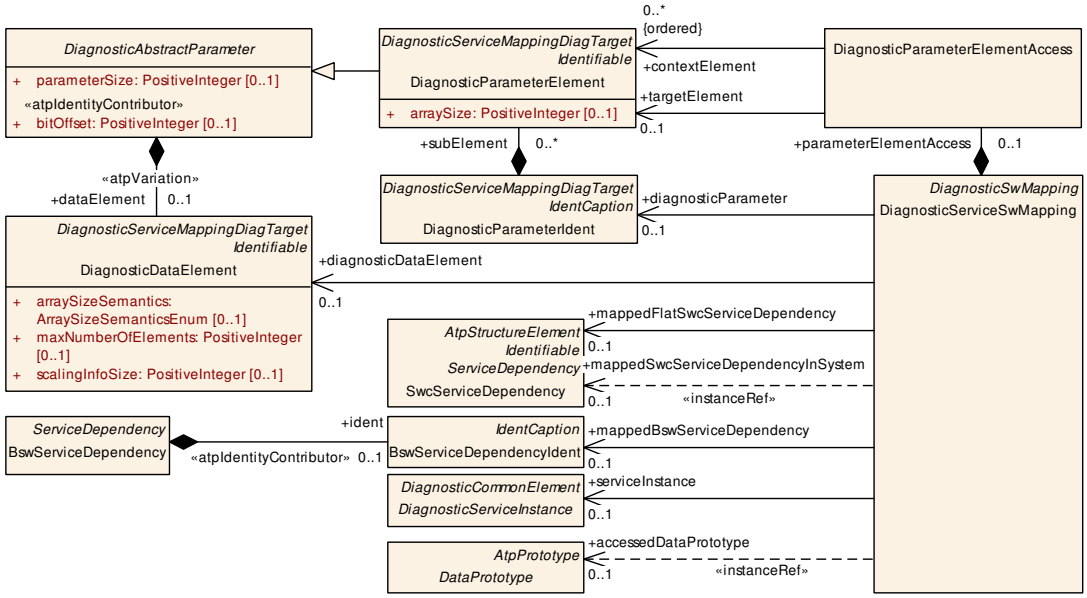

Figure 5.4: Dcm service software mapping

[TPS_DEXT_01181] Existence of references to subclasses of DiagnosticServiceMappingDiagTarget ⌈Within the context of a DiagnosticServiceSwMapping, there are two references and one aggregation available that provide the ability to refer to subclasses of DiagnosticServiceMappingDiagTarget:

diagnosticDataElement This reference is the preferred approach if the mapping affects a primitive DataPrototype or a DataPrototype that represents an array of a primitive AutosarDataType that is mapped to a DiagnosticDataElement that is not a member of a nested DiagnosticParameter.

parameterElementAccess This aggregation is reserved for the case that a DataPrototype is mapped to a level inside the nested definition of a DiagnosticParameter, i.e. if DiagnosticParameter.ident.subElement exists.

In this case, the reference accessedDataPrototype shall also exist in order to define the element in the composite AutosarDataType that corresponds to the referenced parameterElementAccess.targetElement.

diagnosticParameter This option is applicable if a DataPrototype shall be mapped to the root DiagnosticParameter (that defines a nested sub-structure, i.e. DiagnosticParameter.ident.subElement shall exist in this case).

In this case, the structure of the composite DataPrototype shall match the nested structure of the DiagnosticParameter and the DiagnosticParameterElements shall not have any gaps in their placement.

⌋

[TPS_DEXT_01182] Use case for the existence of reference DiagnosticServiceSwMapping.accessedDataPrototype ⌈The reference in the role DiagnosticServiceSwMapping.accessedDataPrototype shall be used to identify a DataPrototype inside an ArgumentDataPrototype that is typed by a composite AutosarDataType such that the referenced DataPrototype – rather than the entire ArgumentDataPrototype – becomes the subject of the mapping to either

- the referenced diagnosticDataElement (i.e. mapping to a flat primitive element of the enclosing DiagnosticAbstractDataIdentifier/DiagnosticRoutineSubfunction) or
- the referenced parameterElementAccess.targetElement (i.e. mapping to an element of a nested structure inside the enclosing DiagnosticAbstractDataIdentifier/DiagnosticRoutineSubfunction).

⌋

Class	***DiagnosticSwMapping*** (abstract)			
Note	This represents the ability to define a mapping between a diagnostic information (at this point there is no way to become more specific about the semantics) to a software-component.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticEdrSenderPortMapping, DiagnosticEdrServerPortMapping, DiagnosticEnableConditionPortMapping, DiagnosticEventPortMapping, DiagnosticFimFunctionMapping, DiagnosticJ1939SwMapping, DiagnosticOperationCyclePortMapping, DiagnosticServiceDataMapping, DiagnosticServiceSwMapping, DiagnosticStorageConditionPortMapping, ISignalPortToDiagnosticEventMapping			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 5.13: DiagnosticSwMapping

Class	**DiagnosticServiceSwMapping**
Note	This represents the ability to define a mapping of a diagnostic service to a software-component or a basic-software module. If the former is used then this kind of service mapping is applicable for the usage of ClientServerInterfaces. **Tags:** atp.recommendedPackage=DiagnosticServiceMappings

▽

△

Class	DiagnosticServiceSwMapping			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, DiagnosticSwMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
accessedData Prototype	DataPrototype	0..1	iref	This instanceRef identifies the DataPrototype that is supposed to be accessed in the context of the operation argument. **Tags:** atp.Splitkey=accessedDataPrototype, accessed DataPrototype.rootDataPrototypeInCs, accessedData Prototype.contextDataPrototypeInCs, accessedData Prototype.targetDataPrototypeInCs **InstanceRef implemented by:** DataPrototypeInClient ServerInterfaceInstanceRef This Attribute is only used by the AUTOSAR Classic Platform.
diagnosticData Element	DiagnosticDataElement	0..1	ref	This represents a DiagnosticDataElement required to execute the respective diagnostic service in the context of the diagnostic service mapping, **Tags:** atp.Splitkey=diagnosticDataElement
diagnostic Parameter	DiagnosticParameter Ident	0..1	ref	This represents the applicable payload that corresponds to the referenced DataPrototype in the role mappedData Element. **Tags:** atp.Splitkey=diagnosticParameter
mappedBsw Service Dependency	BswService DependencyIdent	0..1	ref	This is supposed to represent a reference to a Bsw ServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference. **Tags:** atp.Splitkey=mappedBswServiceDependency
mappedFlatSwc Service Dependency	SwcService Dependency	0..1	ref	This represents the ability to refer to an AtomicSw ComponentType that is available without the definition of how it will be embedded into the component hierarchy. **Tags:** atp.Splitkey=mappedFlatSwcServiceDependency
mappedSwc Service DependencyIn System	SwcService Dependency	0..1	iref	This represents the ability to point into the component hierarchy (under possible consideration of the root SoftwareComposition) **Tags:** atp.Splitkey=mappedSwcServiceDependencyIn System, mappedSwcServiceDependencyInSystem.base, mappedSwcServiceDependencyInSystem.contextRoot SwComposition, mappedSwcServiceDependencyIn System.contextSwComponentPrototype, mappedSwc ServiceDependencyInSystem.targetSwcService Dependency **InstanceRef implemented by:** SwcServiceDependency InSystemInstanceRef
parameter ElementAccess	DiagnosticParameter ElementAccess	0..1	aggr	This aggregation represents the single point of access to the reference to one specific DiagnosticParameter Element. **Tags:** atp.Splitkey=parameterElementAccess
serviceInstance	DiagnosticService Instance	0..1	ref	This represents the service instance that needs to be considered in this diagnostics service mapping. **Tags:** atp.Splitkey=serviceInstance

Table 5.14: DiagnosticServiceSwMapping

[TPS_DEXT_01044] BswServiceDependency needs to act as the target of a reference ⌈The intention of DiagnosticServiceSwMapping.mappedBswServiceDependency is to refer to a BswServiceDependency in the same way as e.g. DiagnosticServiceSwMapping.mappedFlatSwcServiceDependency does.

However, BswServiceDependency is not derived from meta-class Referrable and can therefore never become the target of a reference like DiagnosticServiceSwMapping.mappedBswServiceDependency.

The remedy for this issue is to define meta-class BswServiceDependencyIdent that inherits from IdentCaption that in turn inherits from Referrable.

Then, by aggregating BswServiceDependencyIdent at BswServiceDependency in the role ident BswServiceDependency can **factually** become the target of the reference and thus the original idea of DiagnosticServiceSwMapping.mappedBswServiceDependency becomes feasible.⌋

Please find further details in Figure 5.4.

The introduction [TPS_DEXT_01044], although dangerously close to a workaround, is necessary to keep the AUTOSAR XML Schema fully backwards-compatible.

In other words, if BswServiceDependency were updated to inherit from Referrable, the consequence would be that all existing AUTOSAR models that contain instances BswServiceDependency would suddenly become invalid because Referrable.shortName is a mandatory attribute in the AUTOSAR XML Schema.

Class	**BswServiceDependencyIdent**			
Note	This meta-class is created to add the ability to become the target of a reference to the non-Referrable BswServiceDependency.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *IdentCaption*, *Identifiable*, *Multilanguage Referrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, BswServiceDependency.ident			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table 5.15: BswServiceDependencyIdent

[constr_1346] Allowed values of DiagnosticServiceSwMapping.serviceInstance

Imposition time: CP: IT_DiagDes

⌈The applicability of the DiagnosticServiceSwMapping is limited to a predefined set of diagnostic services.

By regulation of the AUTOSAR standard, DiagnosticServiceSwMapping.serviceInstance shall only point to the following sub-classes of DiagnosticServiceInstance:

- DiagnosticRoutine
- DiagnosticSecurityAccess

- DiagnosticReadDataByIdentifier
- DiagnosticWriteDataByIdentifier
- DiagnosticIOControl

⌋

[constr_1347] Existence of references owned by DiagnosticServiceSwMapping

Imposition time: CP: IT_DiagDes

⌈For any given DiagnosticServiceSwMapping, **one and only one of** the following references shall exist:

- DiagnosticServiceSwMapping.mappedFlatSwcServiceDependency
- DiagnosticServiceSwMapping.mappedSwcServiceDependencyInSystem
- DiagnosticServiceSwMapping.mappedBswServiceDependency

⌋

[constr_1347], among further clarifications, reflects that at most one SwcServiceDependency can be referenced by a DiagnosticServiceSwMapping and this SwcServiceDependency cannot be identified by both mappedSwcServiceDependencyInSystem and mappedFlatSwcServiceDependency.

5.2.4 Diagnostic Secure Coding

The term “secure coding” describes a procedure to write data into the non-volatile memory of the ECU, that is authenticated by the vehicle manufacturer and therefore inhibits write access to unauthorized persons.

Coding data in general contain vehicle configuration information and can be used to activate functions in the ECU software or change its behavior.

The configuration information is persisted and ECU applications dynamically read this information to adapt the behavior to the individual configurations.

To give vehicle manufacturers control over who is writing which data, the ECU shall accept only coding data that is signed with the vehicle manufacturer’s private key. Before secure data is written, a vehicle manufacturer certificate containing a public key is required.

In the next step the configuration data is written into a temporary memory and is only accepted and persisted with a valid signature that matches the certificate’s public key.

[TPS_DEXT_01187] References from DiagnosticSecureCodingMapping to DiagnosticRoutine ⌈The DiagnosticRoutine referenced by meta-class

DiagnosticSecureCodingMapping in the role validation is responsible for finalizing the coding procedure.

The finalization consists mainly of the verification of the procedure and the persistent storage of the coding information after a successful verification.⌋

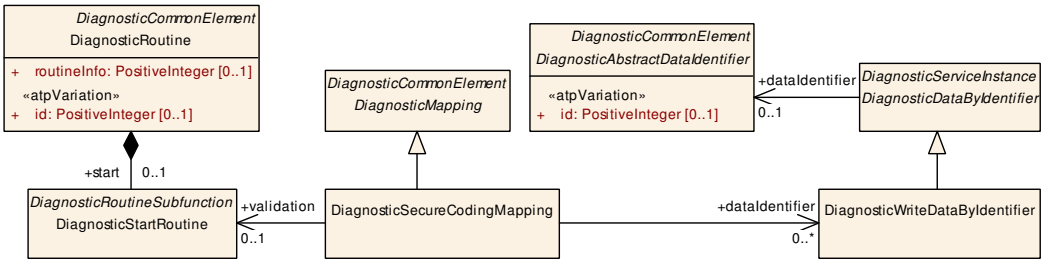

Figure 5.5: Modeling of DiagnosticSecureCodingMapping

Class	**DiagnosticSecureCodingMapping**			
Note	This meta-class acts a mapping element to select diagnostic service instances that are used in the secure coding procedure. Besides, there are already defined diagnostic capabilities they furthermore have a additional semantics that is used during the secure coding process. In particular, this class references write data by identifier instances that mandatory require a secure coding and this class references a diagnostic routine control that is used to provide the signature and to persist the data in non-volatile memory. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataIdentifier	DiagnosticWriteDataByIdentifier	*	ref	Refer to write data by identifier instances that have the semantics to be a secure coding DID. **Tags:** atp.Splitkey=dataIdentifier
validation	DiagnosticStartRoutine	0..1	ref	Refer to a routine control that is used to authenticate and persist the secure coding data. **Tags:** atp.Splitkey=validation

Table 5.16: DiagnosticSecureCodingMapping

5.2.5 Diagnostic Authentication Transmit Certificate Mapping

[TPS_DEXT_01195] Semantics of the DiagnosticAuthTransmitCertificateMapping ⌈Meta-class DiagnosticAuthTransmitCertificateMapping is used to associate the diagnostic service instance for the certificate transmit (i.e., DiagnosticAuthTransmitCertificateEvaluation that is aggregated by DiagnosticAuthTransmitCertificate) with the description of the properties of the corresponding cryptographic certificate, represented by meta-class CryptoServiceCertificate.⌋

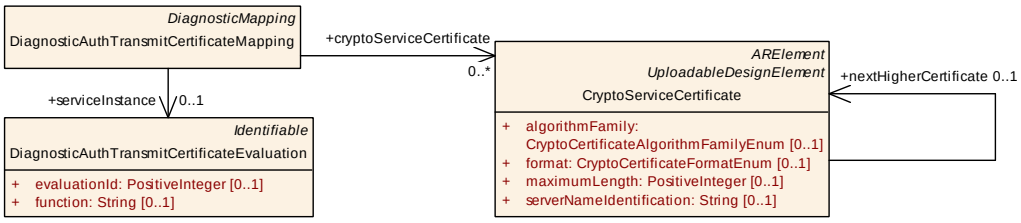

Figure 5.6: Modeling of meta-class `DiagnosticAuthTransmitCertificateMapping`

Class	**DiagnosticAuthTransmitCertificateMapping**			
Note	This meta-class represents the ability to associate a CryptoServiceCertificate with a DiagnosticAuth CertificateEvaluation with the purpose to configure the evaluation of the certificate. **Tags:** atp.recommendedPackage=DiagnosticMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
cryptoService Certificate	CryptoService Certificate	*	ref	This reference identifies the description of the applicable crypto certificate. **Tags:** atp.Splitkey=cryptoServiceCertificate
serviceInstance	DiagnosticAuthTransmit CertificateEvaluation	0..1	ref	This reference identifies the applicable DiagnosticAuth TransmitCertificate service instance (via the aggregation in the role certificateEvaluation). **Tags:** atp.Splitkey=serviceInstance

Table 5.17: DiagnosticAuthTransmitCertificateMapping

5.3 Event-related Diagnostic Mapping

This chapter describes the various mappings (see Figure 5.7) in the context of the description of diagnostic events.

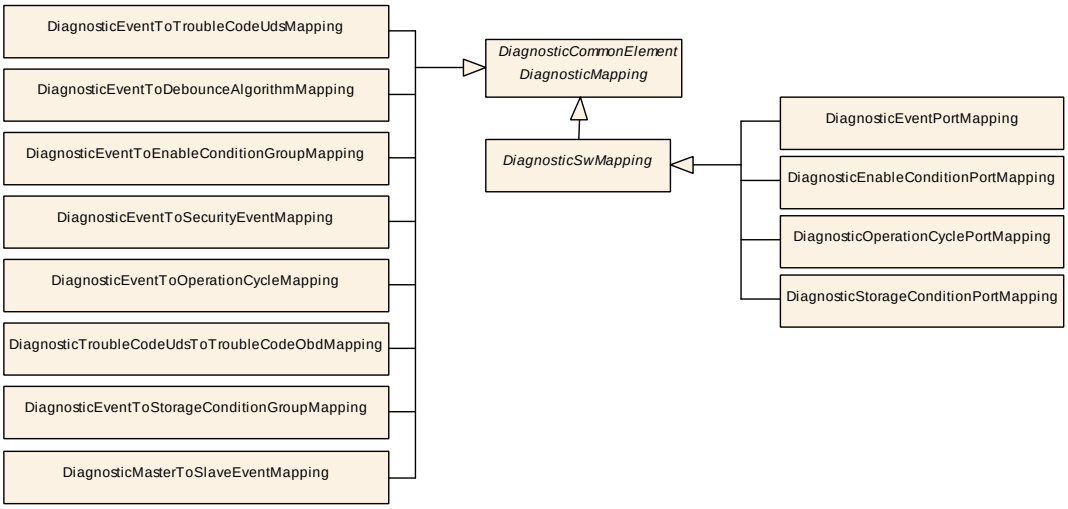

Figure 5.7: Modeling of event-related Diagnostic Mapping

Figure 5.7 gives an overview of the different types of mappings available for diagnostic event handling.

[TPS_DEXT_03002] Two kinds of mappings ⌈For diagnostic event handling, there are two kinds of mappings:

- Mapping between a DiagnosticEvent and another diagnostic resource. For this purpose, a sub-class of DiagnosticMapping is used.
- Mapping between a DiagnosticEvent and a SWC service port. For this purpose, a subclass of DiagnosticSwMapping is used.

⌋

5.3.1 DiagnosticEvent to DtcUds Mapping

[TPS_DEXT_03003] Semantics of DiagnosticEventToTroubleCodeUdsMapping ⌈The DiagnosticEventToTroubleCodeUdsMapping is used to assign one (1:1) or multiple (n:1) DiagnosticEvents to a DiagnosticTroubleCodeUds.

In the case of n:1, multiple instances of DiagnosticEventToTroubleCodeUdsMapping with the same reference of role troubleCodeUds but different references of role diagnosticEvent have to be defined.⌋

[constr_10645] Value of DiagnosticEvent.confirmationThreshold for an n:1 mapping to DiagnosticTroubleCodeUds

Imposition time: CP: IT_DiagDes

⌈If multiple DiagnosticEvents are (by way of multiple DiagnosticEventToTroubleCodeUdsMappings) associated with that same DiagnosticTroubleCodeUds, then the value of attribute confirmationThreshold of **each** of the affected DiagnosticEvents shall be **identical**.⌋

[constr_10646] Value of attributes of DiagnosticConnectedIndicator for an n:1 mapping to DiagnosticTroubleCodeUds where all affected DiagnosticEvents transitively refer to the identical DiagnosticIndicator

Imposition time: CP: IT_DiagDes

⌈If

- multiple DiagnosticEvents are (by way of multiple DiagnosticEventToTroubleCodeUdsMappings) associated with the same DiagnosticTroubleCodeUds **and**
- the target of **all** connectedIndicator.indicator is the identical DiagnosticIndicator,

then the values of the following attributes and references owned by **all** affected DiagnosticConnectedIndicators shall have **the identical value**:

- behavior
- indicatorFailureCycleCounterThreshold
- healingCycleCounterThreshold
- indicator

⌋

[constr_10647] In an n:1 mapping between DiagnosticEvent and DiagnosticTroubleCodeUds, all DiagnosticEvents shall be mapped to the same DiagnosticOperationCycle

Imposition time: CP: IT_DiagDes

⌈If multiple DiagnosticEvents are (by way of multiple DiagnosticEventToTroubleCodeUdsMappings) mapped to the same DiagnosticTroubleCodeUds, then all affected DiagnosticEvents shall (by way of multiple DiagnosticEventToOperationCycleMappings) be mapped to the identical DiagnosticOperationCycle.⌋

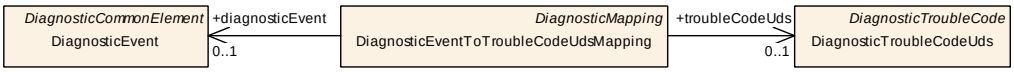

Figure 5.8: DiagnosticEventToDtcUdsMapping

Class	**DiagnosticEventToTroubleCodeUdsMapping**			
Note	Defines which UDS Diagnostic Trouble Code is applicable for a DiagnosticEvent. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to a DiagnosticEvent to which a UDS Diagnostic Trouble Code is assigned. **Tags:** atp.Splitkey=diagnosticEvent
troubleCodeUds	DiagnosticTroubleCodeUds	0..1	ref	Reference to an UDS Diagnostic Trouble Code assigned to a DiagnosticEvent. **Tags:** atp.Splitkey=troubleCodeUds

Table 5.18: DiagnosticEventToTroubleCodeUdsMapping

5.3.2 DiagnosticEvent to DiagnosticOperationCycle Mapping

[TPS_DEXT_01086] Reference to meta-class DiagnosticOperationCycle ⌈A DiagnosticEvent needs to be assigned to exactly one DiagnosticOperationCycle.⌋

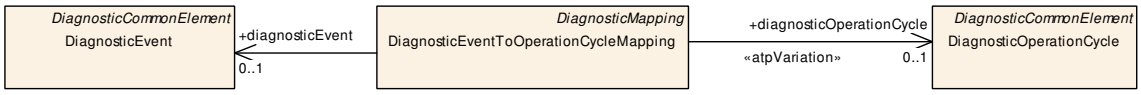

Figure 5.9: DiagnosticEventToOperationCycleMapping

Class	**DiagnosticEventToOperationCycleMapping**			
Note	Defines which OperationCycle is applicable for a DiagnosticEvent. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to a DiagnosticEvent to which an Operation Cycle is assigned. **Tags:** atp.Splitkey=diagnosticEvent
diagnostic OperationCycle	DiagnosticOperation Cycle	0..1	ref	Reference to an OperationCycle assigned to a Diagnostic Event. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=diagnosticOperationCycle, diagnostic OperationCycle.diagnosticOperationCycle, diagnostic OperationCycle.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 5.19: DiagnosticEventToOperationCycleMapping

5.3.3 DiagnosticEvent to DebounceAlgorithm Mapping

[TPS_DEXT_03004] **`DiagnosticEvent` and `DiagnosticDebounceAlgorithmProps`** ⌈If a `DiagnosticEvent` has to be debounced, it shall be mapped to the appropriate `DiagnosticDebounceAlgorithmProps`.⌋

[TPS_DEXT_03005] **Existence of `DiagnosticEventToDebounceAlgorithmMapping`** ⌈The `DiagnosticEventToDebounceAlgorithmMapping` shall not be created if the `DiagnosticEvent` is not debounced.⌋

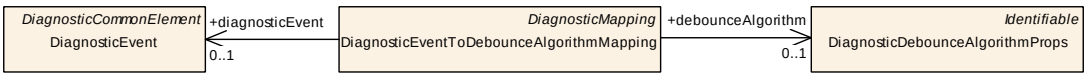

Figure 5.10: DiagnosticEventToDebounceAlgorithmMapping

Class	**DiagnosticEventToDebounceAlgorithmMapping**
Note	Defines which Debounce Algorithm is applicable for a DiagnosticEvent. **Tags:** atp.recommendedPackage=DiagnosticMappings
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

▽

△

Class	**DiagnosticEventToDebounceAlgorithmMapping**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
debounce Algorithm	DiagnosticDebounce AlgorithmProps	0..1	ref	Reference to a DebounceAlgorithm assigned to a DiagnosticEvent. **Tags:** atp.Splitkey=debounceAlgorithm
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to a DiagnosticEvent to which a Debounce Algorithm is assigned. **Tags:** atp.Splitkey=diagnosticEvent

Table 5.20: DiagnosticEventToDebounceAlgorithmMapping

The details regarding the formalization of debouncing behavior are depicted in Figure 4.65.

In particular, DiagnosticCommonProps aggregates DiagnosticDebounceAlgorithmProps in the role debounceAlgorithmProps. The DiagnosticDebounceAlgorithmProps itself does not actually represent the debouncing algorithm but provides attributes relevant for the actual debouncing algorithm.

5.3.4 DiagnosticEvent to EnableConditionGroup Mapping

[TPS_DEXT_03015] EnableConditions have to be put into a DiagnosticEnableConditionGroup ⌈EnableConditions that are assigned to a DiagnosticEvent have to be put into a DiagnosticEnableConditionGroup since only a group of EnableConditions can be mapped to a DiagnosticEvent.⌋

[constr_1361] Number of DiagnosticEventToEnableConditionGroupMapping elements per DiagnosticEvent

Imposition time: CP: IT_DiagDes

⌈The mapping element DiagnosticEventToEnableConditionGroupMapping shall be created no more than once per DiagnosticEvent.

If several DiagnosticEventToEnableConditionGroupMapping elements referring to the same DiagnosticEvent are defined, then the Enable Condition Group mapping shall be regarded as defective.⌋

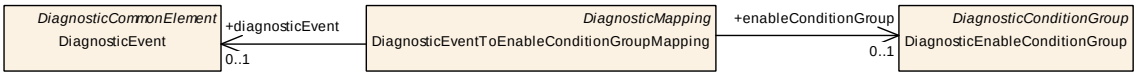

Figure 5.11: DiagnosticEventToEnableConditionGroupMapping

Class	**DiagnosticEventToEnableConditionGroupMapping**			
Note	Defines which EnableConditionGroup is applicable for a DiagnosticEvent. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to a DiagnosticEvent to which an Enable ConditionGroup is assigned. **Tags:** atp.Splitkey=diagnosticEvent
enableCondition Group	DiagnosticEnable ConditionGroup	0..1	ref	Reference to an EnableConditionGroup assigned to a DiagnosticEvent. **Tags:** atp.Splitkey=enableConditionGroup

Table 5.21: DiagnosticEventToEnableConditionGroupMapping

5.3.5 DiagnosticEvent to StorageConditionGroup Mapping

[TPS_DEXT_03016] `StorageCondition`s have to be put into a `DiagnosticStorageConditionGroup` ⌈`StorageCondition`s that are assigned to a `DiagnosticEvent` have to be put into a `DiagnosticStorageConditionGroup` since only a group of `StorageCondition`s can be mapped to a `DiagnosticEvent`.⌋

[constr_1362] Number of `DiagnosticEventToStorageConditionGroupMapping` elements per `DiagnosticEvent`

Imposition time: CP: `IT_DiagDes`

⌈The mapping element `DiagnosticEventToStorageConditionGroupMapping` shall be created no more than once per `DiagnosticEvent`.

If several `DiagnosticEventToStorageConditionGroupMapping` elements referring to the same `DiagnosticEvent` are defined, then the `Storage Condition Group` mapping shall be regarded as defective.⌋

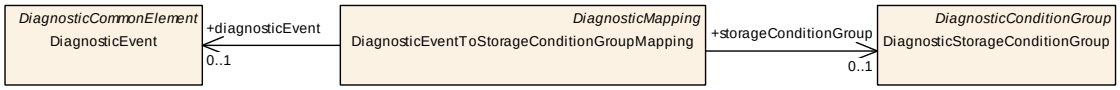

Figure 5.12: DiagnosticEventToStorageConditionGroupMapping

[TPS_DEXT_03006] Values of the individual `DiagnosticStorageCondition`s ⌈The values of the individual `DiagnosticStorageCondition`s need to be algorithmically evaluated in order to find out whether or not the storage of the `DiagnosticEvent` is permitted.

The algorithm that is supposed to be implemented for this purpose is documented in [SWS_Dem_00459].⌋

Class	**DiagnosticEventToStorageConditionGroupMapping**			
Note	Defines which StorageConditionGroup is applicable for a DiagnosticEvent. **Tags:** atp.recommendedPackage=DiagnosticMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to a DiagnosticEvent to which a Storage ConditionGroup is assigned. **Tags:** atp.Splitkey=diagnosticEvent
storage ConditionGroup	DiagnosticStorage ConditionGroup	0..1	ref	Reference to a StorageConditionGroup assigned to a DiagnosticEvent. **Tags:** atp.Splitkey=storageConditionGroup

Table 5.22: DiagnosticEventToStorageConditionGroupMapping

5.3.6 DiagnosticEvent to Port Mapping

[TPS_DEXT_03007] Semantics of `DiagnosticEventPortMapping` ⌈A `DiagnosticEventPortMapping` defines which `SwcServiceDependency`s of a `AtomicSwComponentType` or `BswServiceDependency` of a `BswModuleDescription` have to be connected to which `DiagnosticEvent`.

This is realized by defining a `DiagnosticEventPortMapping` referencing a `DiagnosticEvent` and (using `≪instanceRef≫`) an instance of `SwcServiceDependency` (or `BswServiceDependency`).⌋

If such an instance is not yet available, an ordinary reference to `SwcServiceDependency` can be given alternatively (i.e. without specifying a certain instance).

In this way, the `ECU` integrator is able to directly derive the actual mapping between SWC (or BSW) service ports and the ports of the `Service Components` during `ECU` configuration.

[constr_1435] Debouncing in the presence of a `DiagnosticEventPortMapping`

Imposition time: CP: `IT_DiagDes`

⌈If a `DiagnosticEventPortMapping` exists and the enclosed `DiagnosticEventPortMapping.diagnosticEvent` is also referenced by a `DiagnosticEventToDebounceAlgorithmMapping` then the concrete subclass of the respective `DiagnosticEventToDebounceAlgorithmMapping.debounceAlgorithm.debounceAlgorithm` shall be identical to the `DiagnosticEventPortMapping.swcServiceDependencyInSystem/swcFlatServiceDependency.serviceNeeds.diagEventDebounceAlgorithm`.⌋

For the modeling of `ServiceNeeds`, please refer to [TPS_SWCT_01029].

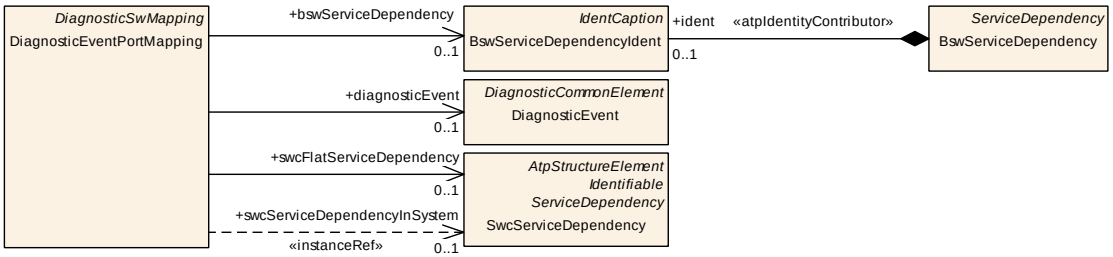

Figure 5.13: DiagnosticEventPortMapping

[constr_10642] Existence of `DiagnosticEventPortMapping.diagnosticEvent` for one specific `DiagnosticEvent`

Imposition time: CP: `IT_DiagDes`

⌈Each `DiagnosticEvent` shall be referenced **at most once** in the role `DiagnosticEventPortMapping.diagnosticEvent`.⌋

In other words, each diagnostic event shall have at most one monitor that reports the status of the event. Several monitors shall not report the status of the same event.

Class	**DiagnosticEventPortMapping**			
Note	Defines to which SWC service ports the DiagnosticEvent is mapped. **Tags:** atp.recommendedPackage=DiagnosticPortMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bswService Dependency	BswService DependencyIdent	0..1	ref	Reference to a BswServiceDependency that links Service Needs to BswModuleEntries. **Tags:** atp.Splitkey=bswServiceDependency This Attribute is only used by the AUTOSAR Classic Platform.
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to the DiagnosticEvent that is assigned to SWC service ports. **Tags:** atp.Splitkey=diagnosticEvent
swcFlatService Dependency	SwcService Dependency	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports. **Tags:** atp.Splitkey=swcFlatServiceDependency This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	**DiagnosticEventPortMapping**			
swcService DependencyIn System	SwcService Dependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports. **Tags:** atp.Splitkey=swcServiceDependencyInSystem, swcServiceDependencyInSystem.base, swcService DependencyInSystem.contextRootSwComposition, swc ServiceDependencyInSystem.contextSwComponent Prototype, swcServiceDependencyInSystem.targetSwc ServiceDependency **InstanceRef implemented by:** SwcServiceDependency InSystemInstanceRef This Attribute is only used by the AUTOSAR Classic Platform.

Table 5.23: DiagnosticEventPortMapping

[constr_1843] Existence of reference `DiagnosticEventPortMapping.diagnosticEvent`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticEventPortMapping`, the reference to `DiagnosticEvent` in the role `diagnosticEvent` shall exist.⌋

[constr_1762] Existence of references owned by `DiagnosticEventPortMapping`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticEventPortMapping`, only one of the references

- to `BswServiceDependency` in the role `bswServiceDependency`
- to `SwcServiceDependency` in the role `swcFlatServiceDependency`
- to `SwcServiceDependency` in the role `swcServiceDependencyInSystem`

shall exist.⌋

5.3.7 DiagnosticOperationCycle to Port Mapping

[TPS_DEXT_03017] Semantics of `DiagnosticOperationCyclePortMapping`

⌈A `DiagnosticOperationCyclePortMapping` defines which SWC service port(s) have to be connected to which `DiagnosticOperationCycle`.

This is realized by defining a `DiagnosticOperationCyclePortMapping` referencing a `DiagnosticOperationCycle` and an instance of `SwcServiceDependency`.⌋

If such an instance is not yet available, an ordinary reference to `SwcServiceDependency` can be given alternatively (i.e. without specifying a certain instance).

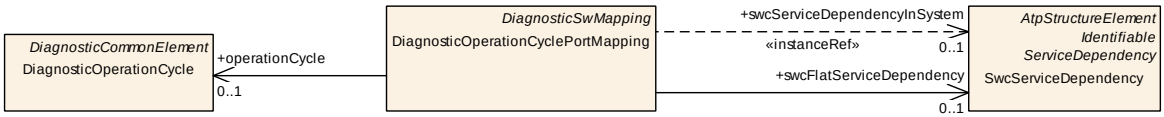

Figure 5.14: DiagnosticOperationCyclePortMapping

Class	**DiagnosticOperationCyclePortMapping**			
Note	Defines to which SWC service ports the DiagnosticOperationCycle is mapped. **Tags:** atp.recommendedPackage=DiagnosticPortMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
operationCycle	DiagnosticOperationCycle	0..1	ref	Reference to the DiagnosticOperationCycle that is assigned to SWC service ports. **Tags:** atp.Splitkey=operationCycle
swcFlatServiceDependency	SwcServiceDependency	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports. **Tags:** atp.Splitkey=swcFlatServiceDependency This Attribute is only used by the AUTOSAR Classic Platform.
swcServiceDependencyInSystem	SwcServiceDependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports. **Tags:** atp.Splitkey=swcServiceDependencyInSystem, swcServiceDependencyInSystem.base, swcServiceDependencyInSystem.contextRootSwComposition, swcServiceDependencyInSystem.contextSwComponentPrototype, swcServiceDependencyInSystem.targetSwcServiceDependency **InstanceRef implemented by:** SwcServiceDependencyInSystemInstanceRef This Attribute is only used by the AUTOSAR Classic Platform.

Table 5.24: DiagnosticOperationCyclePortMapping

[constr_1844] Existence of reference DiagnosticOperationCyclePortMapping.operationCycle

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticOperationCyclePortMapping, the reference to DiagnosticOperationCycle in the role operationCycle shall exist.⌋

[constr_1759] Existence of references owned by DiagnosticOperationCyclePortMapping

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticOperationCyclePortMapping, only one of the following references

- to SwcServiceDependency in the role swcFlatServiceDependency
- to SwcServiceDependency in the role swcServiceDependencyInSystem

shall exist.⌋

5.3.8 DiagnosticEnableCondition to Port Mapping

[TPS_DEXT_03018] Semantics of DiagnosticEnableConditionPortMapping
⌈A DiagnosticEnableConditionPortMapping defines which SWC service port(s) have to be connected to which DiagnosticEnableCondition. This is realized by defining a DiagnosticEnableConditionPortMapping referencing a DiagnosticEnableCondition and an instance of SwcServiceDependency.⌋

If such an instance is not yet available, an ordinary reference to SwcService-Dependency can be given alternatively (i.e. without specifying a certain instance).

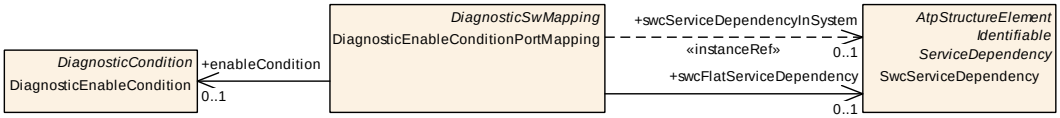

Figure 5.15: DiagnosticEnableConditionPortMapping

[constr_1752] Existence of references owned by meta-class DiagnosticEnableConditionPortMapping

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEnableConditionPortMapping, only one of the following references

- to SwcServiceDependency in the role swcFlatServiceDependency
- to SwcServiceDependency in the role swcServiceDependencyInSystem

may exist.⌋

Class	**DiagnosticEnableConditionPortMapping**			
Note	Defines to which SWC service ports the DiagnosticEnableCondition is mapped. **Tags:** atp.recommendedPackage=DiagnosticPortMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
enableCondition	DiagnosticEnable Condition	0..1	ref	Reference to the EnableCondition which is mapped to a SWC service port. **Tags:** atp.Splitkey=enableCondition
swcFlatService Dependency	SwcService Dependency	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports. This reference can be used in early stages of the development in order to identify the SwcServiceDependency without a full System Context. **Tags:** atp.Splitkey=swcFlatServiceDependency This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	DiagnosticEnableConditionPortMapping			
swcService DependencyIn System	SwcService Dependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports. **Tags:** atp.Splitkey=swcServiceDependencyInSystem, swcServiceDependencyInSystem.base, swcService DependencyInSystem.contextRootSwComposition, swc ServiceDependencyInSystem.contextSwComponent Prototype, swcServiceDependencyInSystem.targetSwc ServiceDependency **InstanceRef implemented by:** SwcServiceDependency InSystemInstanceRef This Attribute is only used by the AUTOSAR Classic Platform.

Table 5.25: DiagnosticEnableConditionPortMapping

[constr_1845] Existence of reference DiagnosticEnableConditionPort-Mapping.enableCondition

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEnableConditionPortMapping, the reference to DiagnosticEnableCondition in the role enableCondition shall exist.⌋

5.3.9 DiagnosticStorageCondition to Port Mapping

[TPS_DEXT_03019] Semantics of DiagnosticStorageConditionPortMapping
⌈A DiagnosticStorageConditionPortMapping defines which SWC service port(s) have to be connected to which DiagnosticStorageCondition. This is realized by defining a DiagnosticStorageConditionPortMapping referencing a DiagnosticStorageCondition and an instance of SwcServiceDependency.⌋

If such an instance is not yet available, an ordinary reference to SwcService-Dependency can be given alternatively (i.e. without specifying a certain instance).

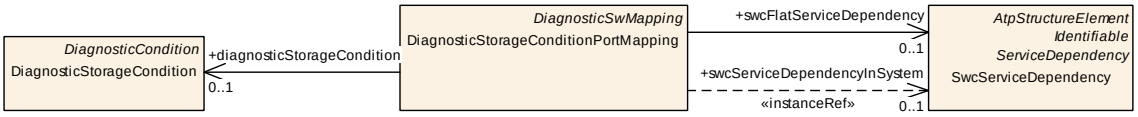

Figure 5.16: DiagnosticStorageConditionPortMapping

[constr_1753] Existence of references owned by meta-class DiagnosticStorageConditionPortMapping

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticStorageConditionPortMapping, only one of the following references

- to SwcServiceDependency in the role swcFlatServiceDependency
- to SwcServiceDependency in the role swcServiceDependencyInSystem

may exist.⌋

Class	DiagnosticStorageConditionPortMapping			
Note	Defines to which SWC service ports with DiagnosticStorageConditionNeeds the DiagnosticStorage Condition is mapped. **Tags:** atp.recommendedPackage=DiagnosticMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnostic Storage Condition	DiagnosticStorage Condition	0..1	ref	Reference to the StorageCondition which is mapped to a SWC service port with DiagnosticStorageCondition Needs. **Tags:** atp.Splitkey=diagnosticStorageCondition
swcFlatService Dependency	SwcService Dependency	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports. **Tags:** atp.Splitkey=swcFlatServiceDependency
swcService DependencyIn System	SwcService Dependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports. **Tags:** atp.Splitkey=swcServiceDependencyInSystem, swcServiceDependencyInSystem.base, swcService DependencyInSystem.contextRootSwComposition, swc ServiceDependencyInSystem.contextSwComponent Prototype, swcServiceDependencyInSystem.targetSwc ServiceDependency **InstanceRef implemented by:** SwcServiceDependency InSystemInstanceRef

Table 5.26: DiagnosticStorageConditionPortMapping

[constr_1846] Existence of reference `DiagnosticStorageConditionPortMapping.diagnosticStorageCondition`

Imposition time: CP: `IT_DiagDes`

⌈For each `DiagnosticStorageConditionPortMapping`, the reference to `DiagnosticStorageCondition` in the role `diagnosticStorageCondition` shall exist.⌋

5.3.10 Extended Data Record Mapping

The values contained in an extended data record can be obtained from different sources:

- Internal values of the diagnostic stack can be assigned by means of the `DiagnosticEdrDataProviderMapping`. This aspect is described in Chapter 5.3.10.1.
- Values from the application software can be assigned by means of the `DiagnosticEdrSenderPortMapping`. This aspect is described in Chapter 5.3.10.2.

- Values from the application software can be assigned by means of the DiagnosticEdrServerPortMapping. This aspect is described in Chapter 5.3.10.3.

5.3.10.1 Extended Data Record Data Provider Mapping

[TPS_DEXT_03020] Semantics of DiagnosticEdrDataProviderMapping ⌈The meta-class DiagnosticEdrDataProviderMapping does not seem to fulfill the condition for representing a mapping class because it only has one reference to a DiagnosticExtendedDataRecordElement in the role extendedDataRecordElement.

However, the specific nature of this mapping is that the second element (the DiagnosticEdrDataProviderMapping.dataProvider) that is supposed to take place in the mapping cannot precisely be modeled as a single meta-class.

Therefore, there is no better way than to model the DiagnosticEdrDataProviderMapping.dataProvider by a NameToken.

The semantics of this mapping is to further qualify the access to the DiagnosticExtendedDataRecordElement referenced in the role extendedDataRecordElement from within the Dem.⌋

The existence of a DiagnosticEdrDataProviderMapping implies that only a very limited number of possibilities (as listed in [TPS_DEXT_01142]) exist to fill a DiagnosticExtendedDataRecordElement with data.

Each of the possible candidates directly implies information about the size of the data, the data type and the possible conversions between internal values and textual labels.

In addition, the existence of a DiagnosticEdrDataProviderMapping is resolved internally inside the diagnostic server.

That said, it is very likely that, from a methodological point of view, the definition of the DiagnosticExtendedDataRecord and DiagnosticEdrDataProviderMapping are created in more or less the same design step.

This strong link between the definition of an DiagnosticExtendedDataRecord and the definition of how to fill the content with local information for the diagnostic server justifies a slightly shortened configuration workflow.

Specifically, it does not make much sense to define an DiagnosticExtendedDataRecordElement along with a detailed representation as SwDataDefProps when the corresponding DiagnosticEdrDataProviderMapping is created in practically the same design step.

Therefore, the existence of DiagnosticExtendedDataRecordElement.swDataDefProps is not mandatory if a DiagnosticEdrDataProviderMapping exists that refers to the DiagnosticExtendedDataRecordElement.

[TPS_DEXT_01203] Existence of DiagnosticExtendedDataRecordElement.swDataDefProps ⌈If a DiagnosticEdrDataProviderMapping exists that refers to a DiagnosticExtendedDataRecordElement, then the reference in the role DiagnosticExtendedDataRecordElement.swDataDefProps is not required.

If the reference exists, then the relevant attributes baseType and compuMethod shall match the requirements imposed by the specific DiagnosticEdrDataProvider-Mapping.dataProvider.⌋

[TPS_DEXT_01142] Standardized values for attribute DiagnosticEdrData-ProviderMapping.dataProvider ⌈

Standardized value	Description
DEM_AGINGCTR_DOWNCNT	Map down-counting Dem-internal aging counter.
DEM_AGINGCTR_UPCNT	Map up-counting Dem-internal aging counter.
DEM_AGINGCTR_UPCNT_FIRST_ACTIVE	Map up-counting Dem-internal aging counter, starting at 1 when aging starts.
DEM_CLR_DIST	Distance since diagnostic trouble codes cleared.
DEM_CLR_TIME	Time since diagnostic trouble codes cleared.
DEM_CURRENT_FDC	Map Dem-internal fault detection counter.
DEM_CYCLES_TESTED_SINCE_FIRST_FAILED	Map Dem-internal Operation Cycle Counter - Tested cycles since first failed excluding test not complete, max. range: 1 byte.
DEM_CYCLES_SINCE_FIRST_FAILED	Map Dem-internal Operation Cycle Counter - cycles since first failed.
DEM_CYCLES_TESTED_SINCE_LAST_FAILED	Map Dem-internal Operation Cycle Counter - Tested cycles since last failed excluding test not complete, max. range: 1 byte.
DEM_CYCLES_SINCE_LAST_FAILED	Map Dem-internal Operation Cycle Counter - cycles since last failed.
DEM_DTC_PRIORITY	DTC priority statically assigned to this DTC.
DEM_FAILED_CYCLES	Map Dem-internal Operation Cycle Counter - failed cycles.
DEM_MAX_FDC_DURING_CURRENT_CYCLE	Map Dem-internal DTC Fault Detection Counter maximum value during current operation cycle.
DEM_MAX_FDC_SINCE_LAST_CLEAR	Map Dem-internal DTC Fault Detection Counter maximum value since last clear.
DEM_MIL_DIST	Distance traveled While MIL is activated.
DEM_MIL_TIME	Time run by the engine while MIL is activated.
DEM_OCCCTR	Map Dem-internal occurrence counter.
DEM_OVFLIND	Map Dem-internal overflow indication.
DEM_SIGNIFICANCE	Map (static) Dem-internal event significance.
DEM_WARM_UPS	Number of warm-ups since diagnostic trouble codes cleared.
DEM_J1939LAMP_STATUS	Current state of applicable lamp(s) for this DTC in SAE J1939 DM31 format.
DEM_EVENT_ASSOCIATED_IDENTIFICATION	Static event identification value associated to an event, see [TPS_DEXT_01151].
IN_USE_MONITOR_PERFORMANCE_RATIO	IUMPR ratio according to [22, SAE J1979-2].
DIAGNOSTIC_TEST_RESULT	DTR test values according to [22, SAE J1979-2].
MONITOR_ACTIVITY_DATA	Monitor activity data according to [22, SAE J1979-2].

▽

△

DEM_OBDDTC_2BYTE	Map OBD number statically assigned to DemObdDTC. DemDtcValue.
DEM_OBDDTC_3BYTE	Map OBD number statically assigned to DemObdDTC. DemDtcValue3Byte.

⌋

Where applicable, AUTOSAR allows for the usage of values of the attribute DiagnosticEdrDataProviderMapping.dataProvider other than the values standardized by [TPS_DEXT_01142].

In this case, however, additional proprietary values of the attribute DiagnosticEdr-DataProviderMapping.dataProvider shall be prefixed with a company-specific name fragment in order to avoid collisions that could occur if or when the list of possible values claimed by the AUTOSAR standard itself is extended.

Please note the existence of [constr_1756].

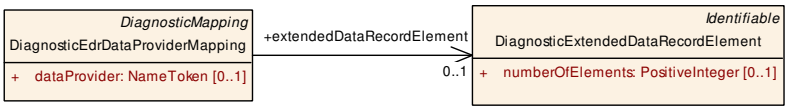

Figure 5.17: Modeling of the DiagnosticEdrDataProviderMapping

Class	**DiagnosticEdrDataProviderMapping**			
Note	This meta-class represents the definition of a data provider for the content of an extended data record. **Tags:** atp.recommendedPackage=DiagnosticDataProviders			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataProvider	NameToken	0..1	attr	This attribute defines the kind of data that is associated with the referenced DiagnosticExtendedDataRecord Element.
extendedData RecordElement	DiagnosticExtended DataRecordElement	0..1	ref	This reference identifies the DiagnosticExtendedData RecordElement to which the data provider is associated with. **Tags:** atp.Splitkey=extendedDataRecordElement

Table 5.27: DiagnosticEdrDataProviderMapping

[constr_1847] Existence of reference DiagnosticEdrDataProviderMapping.extendedDataRecordElement

Imposition time: AP: IT_DiagDes, CP: IT_DiagDes

⌈For each DiagnosticEdrDataProviderMapping, the reference to DiagnosticExtendedDataRecordElement in the role extendedDataRecord-Element shall exist.⌋

5.3.10.2 Extended Data Record Sender Port Mapping

[TPS_DEXT_01197] Semantics of the DiagnosticEdrSenderPortMapping
⌈Meta-class DiagnosticEdrSenderPortMapping is used to configure a connection between a DataPrototype defined in a PortInterface that is used to type a specific PortPrototype and a DiagnosticExtendedDataRecordElement.

The meaning of this mapping is that the value of the DiagnosticExtendedDataRecordElement is sourced from the value of the mapped DataPrototype.⌋



Figure 5.18: Modeling of the DiagnosticEdrSenderPortMapping

Class	**DiagnosticEdrSenderPortMapping**			
Note	This class is used to map a primitive DataPrototype owned by a PortPrototype that is typed by a Data Interface to an element of an extended data record. **Tags:** atp.recommendedPackage=DiagnosticPortMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataPrototype	DataPrototype	0..1	iref	This reference identifies the DataPrototype that shall be accessed for filling the extend data record in the context of a System. **Tags:** atp.Splitkey=dataPrototype, dataPrototype.context RootComposition, dataPrototype.contextComponent, dataPrototype.contextPort, dataPrototype.rootData Prototype, dataPrototype.contextDataPrototype, data Prototype.targetDataPrototype **InstanceRef implemented by:** DataPrototypeInSystem InstanceRef
recordElement	DiagnosticExtended DataRecordElement	0..1	ref	This reference identifies the applicable extended data record element. **Tags:** atp.Splitkey=recordElement

Table 5.28: DiagnosticEdrSenderPortMapping

[constr_10678] Existence of the role DiagnosticEdrSenderPortMapping.dataPrototype

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEdrSenderPortMapping, the reference in the role DiagnosticEdrSenderPortMapping.dataPrototype shall exist.⌋

[constr_10679] Existence of DiagnosticEdrSenderPortMapping.recordElement

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEdrSenderPortMapping, the reference in the role DiagnosticEdrSenderPortMapping.recordElement shall exist.⌋

5.3.10.3 Extended Data Record ServerPort Mapping

[TPS_DEXT_01198] Semantics of the `DiagnosticEdrServerPortMapping`
⌈Meta-class `DiagnosticEdrServerPortMapping` is used to configure a connection between **either**

- a `SwcServiceDependency` defined in an `AtomicSwComponentType` **or**
- a `BswServiceDependency` defined in a `BswModuleDescription`

and a `DiagnosticExtendedDataRecordElement`.⌋

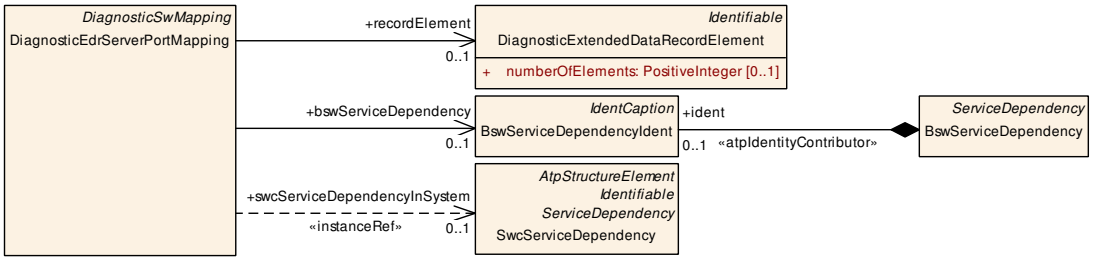

Figure 5.19: Modeling of the `DiagnosticEdrServerPortMapping`

Class	**DiagnosticEdrServerPortMapping**			
Note	This mapping class defines to which SWC service ports the element of the affected extended data record is mapped. **Tags:** atp.recommendedPackage=DiagnosticPortMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bswService Dependency	BswService DependencyIdent	0..1	ref	Reference to a BswServiceDependency that links Service Needs to BswModuleEntries. **Tags:** atp.Splitkey=bswServiceDependency
recordElement	DiagnosticExtended DataRecordElement	0..1	ref	This reference identifies the element of the extended data record to which the value obteained from either application software or basic software shall be assigned. **Tags:** atp.Splitkey=recordElement
swcService DependencyIn System	SwcService Dependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports. **Tags:** atp.Splitkey=swcServiceDependencyInSystem, swcServiceDependencyInSystem.base, swcService DependencyInSystem.contextRootSwComposition, swc ServiceDependencyInSystem.contextSwComponent Prototype, swcServiceDependencyInSystem.targetSwc ServiceDependency **InstanceRef implemented by:** SwcServiceDependency InSystemInstanceRef

Table 5.29: DiagnosticEdrServerPortMapping

[constr_10680] Existence of role DiagnosticEdrServerPortMapping.swcServiceDependencyInSystem vs. bswServiceDependency

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEdrServerPortMapping, **either** the reference in the role

- DiagnosticEdrServerPortMapping.swcServiceDependencyInSystem **or**
- DiagnosticEdrServerPortMapping.bswServiceDependency

shall exist.⌋

[constr_10681] Existence of the role DiagnosticEdrServerPortMapping.recordElement

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEdrServerPortMapping, the reference in the role DiagnosticEdrServerPortMapping.recordElement shall exist.⌋

5.3.11 Master to Slave Event Mapping

AUTOSAR provides the ability to associate multiple diagnostic events such that the reporting of one event (also known as the master event) from a diagnostic monitor automatically triggers the reporting of one or more other diagnostic events (also known as slave events) with the same test result.

The existence of the DiagnosticMasterToSlaveEventMapping is primarily motivated by the need to store the result of event reporting simultaneously in different fault memories within the affected ECU.

As a consequence, a model that references a given DiagnosticEvent as both masterEvent and slaveEvent shall be considered broken, as further clarified in [constr_1590].

In the same spirit, a model where one specific DiagnosticEvent is referenced more than once in the role masterEvent or the role slaveEvent is also considered broken.

[constr_1590] DiagnosticEvent referenced in the role masterEvent or slaveEvent

Imposition time: CP: IT_DiagDes

⌈Any given DiagnosticEvent shall be referenced at most once from a DiagnosticMasterToSlaveEventMapping.⌋

A diagnostic slave event shall never be reported directly by a diagnostic monitor. Slave events are exclusively reported internally in the Dem. This important aspect is covered by [constr_1591].

[constr_1591] `DiagnosticEvent` referenced as `slaveEvent` shall not be reported by diagnostic monitor

Imposition time: CP: `IT_DiagDes`

⌈Any `DiagnosticEvent` that is referenced in the role `DiagnosticMasterToSlaveEventMapping.slaveEvent` shall not be referenced in the role `DiagnosticEventPortMapping.diagnosticEvent` and vice versa.⌋

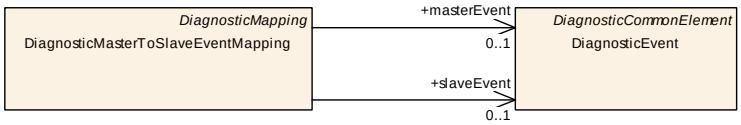

Figure 5.20: Modeling of the `DiagnosticMasterToSlaveEventMapping`

Class	**DiagnosticMasterToSlaveEventMapping**			
Note	This meta-class provides the ability to map a master diagnostic event with a slave diagnostic event such that reporting of the master event with a given value also reports the slave event with the same value **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
masterEvent	DiagnosticEvent	0..1	ref	This represents the master diagnostic event. **Tags:** atp.Splitkey=masterEvent
slaveEvent	DiagnosticEvent	0..1	ref	This represents the slave diagnostic event. **Tags:** atp.Splitkey=slaveEvent

Table 5.30: DiagnosticMasterToSlaveEventMapping

5.3.12 Diagnostic Event to Security Event Mapping

AUTOSAR provides the ability to define so-called security events (formalized by meta-class `SecurityEventDefinition`) to indicate a possible intrusion into an AUTOSAR system.

The details are described in the specification of the [23, TPS Security Extract Template].

[TPS_DEXT_01153] Semantics of meta-class `DiagnosticEventToSecurityEventMapping`

Status: DRAFT

⌈The occurrence of some security events needs to be persisted in the context of the Dem.

Technically, this mechanism boils down to a reporting of a diagnostic event in response to the occurrence of a security event.

This approach is made possible by the existence of the DiagnosticEventTo-SecurityEventMapping.⌋

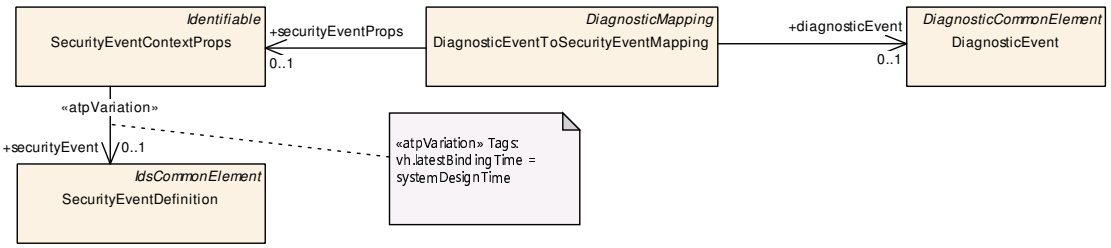

Figure 5.21: Modeling of the DiagnosticEventToSecurityEventMapping

Class	**DiagnosticEventToSecurityEventMapping**			
Note	This meta-class represents the ability to map a security event that is defined in the context of the Security Extract to a diagnostic event defined on the context of the DiagnosticExtract. **Tags:** atp.Status=candidate atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnosticEvent	DiagnosticEvent	0..1	ref	This reference identifies the applicable diagnostic event. **Tags:** atp.Splitkey=diagnosticEvent atp.Status=candidate This Attribute is only used by the AUTOSAR Classic Platform.
securityEvent Props	SecurityEventContext Props	0..1	ref	This reference identifies the qualification of the applicable security event **Tags:** atp.Splitkey=securityEventProps atp.Status=candidate This Attribute is only used by the AUTOSAR Classic Platform.

Table 5.31: DiagnosticEventToSecurityEventMapping

[constr_10026] Existence of the reference in the role DiagnosticEventTo-SecurityEventMapping.diagnosticEvent

Status: DRAFT
Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticEventToSecurityEventMapping, the reference to DiagnosticEvent in the role diagnosticEvent shall exist.⌋

[constr_10027] Existence of the reference in the role DiagnosticEventTo-SecurityEventMapping.securityEventProps

Status: DRAFT
Imposition time: CP: IT_DiagDes, AP: IT_DiagDes

⌈For each DiagnosticEventToSecurityEventMapping, the reference to SecurityEventContextProps in the role securityEventProps shall exist.⌋

It is important to understand that the mapping to a security event is **not** directed at SecurityEventDefinition. This meta-class represents a high-level design object that needs some more context in order to be considered in a specific project.

This additional context is provided by SecurityEventContextProps, which consequently becomes the target of the reference in the role DiagnosticEventToSecurityEventMapping.securityEventProps.

At first glance, it may seem that the definition of the DiagnosticEventToSecurityEventMapping and the existence of attribute SecurityEventContextProps.persistentStorage create a certain level of redundancy. However, this is not really the case.

This means in particular that the existence of SecurityEventContextProps.persistentStorage can be used to indicate the need to persist the respective security event.

The corresponding DiagnosticEventToSecurityEventMapping may be created later in the workflow where the formal information about which security events shall be persisted represents an important input to the workflow.

Class	**SecurityEventContextProps**			
Note	This meta-class specifies the SecurityEventDefinition to be mapped to an IdsmInstance and adds mapping-dependent properties of this security event valid only for this specific mapping.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*SecurityEventContextMapping*.mappedSecurityEvent			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
default ReportingMode	SecurityEventReporting ModeEnum	0..1	attr	This attribute defines the default reporting mode for the referenced security event.
persistent Storage	Boolean	0..1	attr	This attribute controls whether qualified reportings of the referenced security event shall be stored persistently by the mapped IdsmInstance or not.
securityEvent	SecurityEventDefinition	0..1	ref	This reference defines the security event that is mapped and enriched by SecurityEventMappingProps with mapping dependent properties. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=securityEvent, securityEvent.securityEvent Definition, securityEvent.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
sensorInstance Id	PositiveInteger	0..1	attr	This attribute defines the ID of the security sensor that detects the referenced security event.
severity	PositiveInteger	0..1	attr	This attribute defines how critical/severe the referenced security event is. Please note that currently, the severity level meanings of specific integer values is not specified by AUTOSAR but left to the party responsible for the IDS system design (e.g. the OEM).

Table 5.32: SecurityEventContextProps

Class	SecurityEventDefinition			
Note	This meta-class defines a security-related event as part of the intrusion detection system. **Tags:** atp.recommendedPackage=SecurityEventDefinitions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
eventSymbol Name	SymbolProps	0..1	aggr	This aggregation defines optionally an alternative Event Name for the SecurityEventDefinition in case there is a collision of shortNames. **Tags:** atp.Splitkey=eventSymbolName, eventSymbol Name.shortName
id	PositiveInteger	0..1	attr	This attribute represents the numerical identification of the defined security event. The identification shall be unique within the scope of the IDS.
securityEvent ContextData Definition	SecurityEventContext DataDefinition	*	ref	Definition of additional context data that is reported with the security event in order to better support the analysis of a possible security threat. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=securityEventContextDataDefinition, security EventContextDataDefinition.securityEventContextData Definition, securityEventContextDataDefinition.variation Point.shortLabel vh.latestBindingTime=systemDesignTime

Table 5.33: SecurityEventDefinition

Class	SecurityEventContextDataDefinition			
Note	This meta-class represents the possibility to add context data to the referencing SecurityEventDefinition. **Tags:** atp.recommendedPackage=SecurityEventContextDataDefinitions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
contextData Element (ordered)	SecurityEventContext DataElement	*	aggr	Description of contained SecurityEventContextData Elements. If the SecurityEventContextDataDefintion has a primitive type then only one SecurityEventContextData Element shall be used. If the SecurityEventContextData Defintion is structured into several elements then for each one a SecurityEventContextDataElement shall be aggregated. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=systemDesignTime
version	PositiveInteger	0..1	attr	Version number of the context data. For more details see the IDSM protocol specfication.

Table 5.34: SecurityEventContextDataDefinition

5.3.13 Diagnostic Event to ISignalPort Mapping

[TPS_DEXT_01199] Semantics of meta-class ISignalPortToDiagnosticEventMapping

Status: DRAFT

⌈The ISignalPortToDiagnosticEventMapping defines all DiagnosticEvents that are reported if, during a reception of an ISignal or ISignalGroup on an ISignalPort an error is detected.⌋

An ISignalPortToDiagnosticEventMapping may contain several references to DiagnosticEvents in the different roles at the same time.

[TPS_DEXT_01200] Semantics of the reference in the role ISignalPortToDiagnosticEventMapping.iSignalTriggering

Status: DRAFT

⌈The reference in the role ISignalPortToDiagnosticEventMapping.iSignalTriggering defines the ISignalTriggering on a PhysicalChannel of an ISignal or ISignalGroup that is received by an EcuInstance via the iSignalPort.⌋

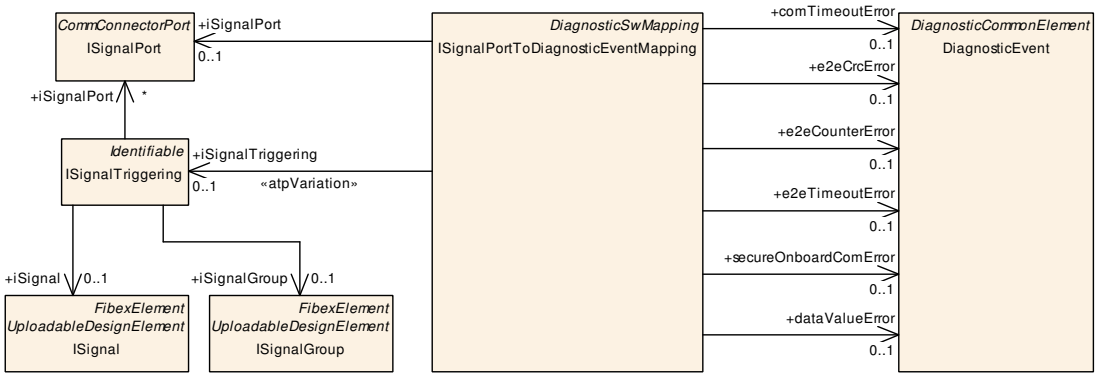

Figure 5.22: Modeling of the ISignalPortToDiagnosticEventMapping

[constr_10687] Existence of the reference ISignalPortToDiagnosticEventMapping.iSignalTriggering

Status: DRAFT
Imposition time: CP: IT_DiagDes

⌈For each ISignalPortToDiagnosticEventMapping, the reference to meta-class ISignalTriggering in the role iSignalTriggering shall exist.⌋

[constr_10688] Existence of the reference ISignalPortToDiagnosticEventMapping.iSignalPort

Status: DRAFT
Imposition time: CP: IT_DiagDes

⌈For each ISignalPortToDiagnosticEventMapping, the reference to meta-class ISignalPort in the role iSignalPort shall exist.⌋

[constr_10689] Restriction for the attribute `communicationDirection` of an `ISignalPort` that is referenced by `ISignalPortToDiagnosticEventMapping`

Status: DRAFT
Imposition time: CP: `IT_DiagDes`

⌈Each `ISignalPort` that is referenced in the role `iSignalPort` by a `ISignalPortToDiagnosticEventMapping` shall set the attribute `communicationDirection` to the value `CommunicationDirectionType.in`.⌋

Class	**ISignalPortToDiagnosticEventMapping**			
Note	This mapping is used to identify the DiagnosticEvents that are reported if during the reception of an ISignalGroup on a specific Ecu an error is detected. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
comTimeout Error	DiagnosticEvent	0..1	ref	Reference to a diagnostic event that is used if a com timeout error (ISignalGroup is not received within expected time) occurred. **Tags:** atp.Splitkey=comTimeoutError xml.sequenceOffset=10
dataValueError	DiagnosticEvent	0..1	ref	Reference to a diagnostic event that is used if a data value error (invalid or out of range) occurred. **Tags:** atp.Splitkey=dataValueError xml.sequenceOffset=60
e2eCounter Error	DiagnosticEvent	0..1	ref	Reference to a diagnostic event that is used if an E2E CRC counter error occurred. **Tags:** atp.Splitkey=e2eCounterError xml.sequenceOffset=30
e2eCrcError	DiagnosticEvent	0..1	ref	Reference to a diagnostic event that is used if an E2E CRC error occurred. **Tags:** atp.Splitkey=e2eCrcError xml.sequenceOffset=20
e2eTimeout Error	DiagnosticEvent	0..1	ref	Reference to a diagnostic event that is used if an E2E timeout error occurred. **Tags:** atp.Splitkey=e2eTimeoutError xml.sequenceOffset=40
iSignalPort	ISignalPort	0..1	ref	Reference to an ISignalPort on which the ISignalGroup is received. **Tags:** atp.Splitkey=iSignalPort xml.sequenceOffset=110

▽

△

Class	ISignalPortToDiagnosticEventMapping			
iSignal Triggering	ISignalTriggering	0..1	ref	Reference to an ISignalTriggering of an ISignalGroup that is received on the ISignalPort. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iSignalTriggering, iSignalTriggering.iSignal Triggering, iSignalTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild xml.sequenceOffset=100
secureOnboard ComError	DiagnosticEvent	0..1	ref	Reference to a diagnostic event that is used if a SecOc error occurred. **Tags:** atp.Splitkey=secureOnboardComError xml.sequenceOffset=50

Table 5.35: ISignalPortToDiagnosticEventMapping

5.4 Mapping of Function Inhibition

The relevant mappings in the context of function inhibition are depicted in Figure 5.23.

The description of the modeling of the mapping in the context of function inhibition can be roughly divided into the following sections:

- The mapping between an inhibition source and a diagnostic event is described in Chapter 5.4.1.
- The alias event mapping is described in Chapter 5.4.2.
- The mapping between a function identifier and a diagnostic monitor is described in Chapter 5.4.3.
- The mapping between a function identifier and an `IUMPR` is described in Chapter 5.4.4.

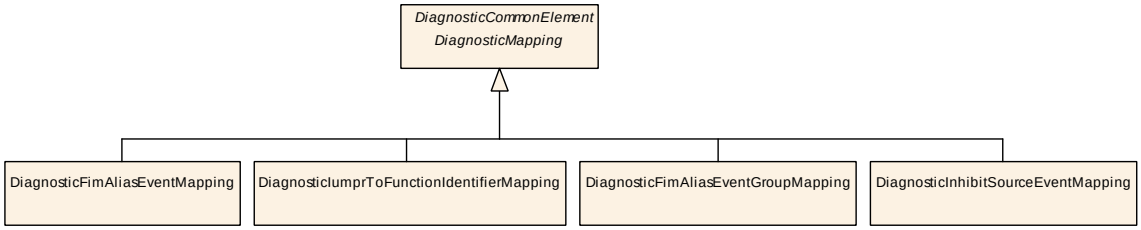

Figure 5.23: Modeling of the `DiagnosticMapping` in the context of function inhibition

5.4.1 Mapping between Inhibition Source and Diagnostic Event

The configuration of the Fim requires the clarification of the relation between a particular inhibition source (modeled as `DiagnosticFunctionInhibitSource`) and one or more diagnostic events (modeled as `DiagnosticEvent`).

The easiest way to provide this information is to use the DiagnosticInhibitSourceEventMapping that is able to create an association between a DiagnosticFunctionInhibitSource on the one hand and either a DiagnosticEvent or a DiagnosticFimEventGroup on the other hand.

This is only possible if the DiagnosticEvents referenced by the DiagnosticInhibitSourceEventMapping already exist. This existence is subject to the development workflow and therefore may or may not apply. For more details, please refer to Figure 4.74.

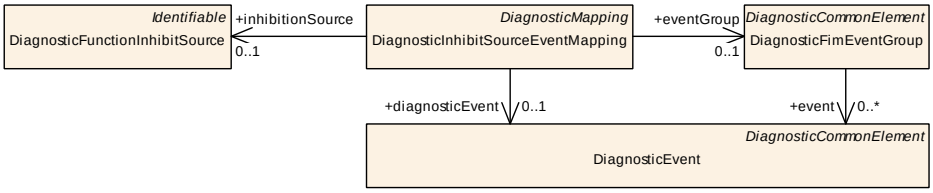

Figure 5.24: Modeling of the DiagnosticInhibitSourceEventMapping

Class	**DiagnosticInhibitSourceEventMapping**			
Note	This meta-class represents the ability to map a DiagnosticFunctionInhibitSource directly to alternatively one DiagnosticEvent or one DiagnosticFimSummaryEvent. This model element shall be used if the approach via the alias events is not applicable, i.e. when diagnostic events defined by the Dem are already available at the time the Fim configuration within the diagnostic extract is created. **Tags:** atp.recommendedPackage=DiagnosticInhibitSourceEventMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnosticEvent	DiagnosticEvent	0..1	ref	This represents the reference to the diagnostic event. **Tags:** atp.Splitkey=diagnosticEvent
eventGroup	DiagnosticFimEvent Group	0..1	ref	This represents the reference to the event group **Tags:** atp.Splitkey=eventGroup
inhibitionSource	DiagnosticFunction InhibitSource	0..1	ref	This represents the reference to the inhibition source. **Tags:** atp.Splitkey=inhibitionSource

Table 5.36: DiagnosticInhibitSourceEventMapping

5.4.2 Alias Event Mapping

[TPS_DEXT_01095] Definition of "alias" diagnostic event for the creation of a Fim configuration in the diagnostic extract ⌈A pre-configuration of the Fim function inhibition can be created on the basis of the following meta-classes:

DiagnosticFimAliasEventMapping in this case the definition of "alias" diagnostic event corresponds to a single DiagnosticEvent

DiagnosticFimAliasEventGroupMapping in this case the definition of a group of "alias" diagnostic events corresponds to a group of single DiagnosticEvents

⌋

Ultimately, the modeling approach for the Fim starts with the definition of the concept of a function itself by means of the meta-class DiagnosticFunctionIdentifier.

[TPS_DEXT_01100] Consequence of the existence of DiagnosticFimAliasEventMapping ⌈The references from meta-class DiagnosticFimAliasEventMapping in the roles actualEvent and aliasEvent do not necessarily have to exist at the same time. The following rules apply:

Only actualEvent exists: In this case there is no DiagnosticFimAliasEvent available and the actual DiagnosticEvent can be taken directly for creating the Fim configuration.

Only aliasEvent exists: In this case the configuration is considered incomplete and only defines the intended semantics from the point of view of a pre-configuration of the Fim.

Both actualEvent and aliasEvent exist: In this case it is obvious that the configuration has undergone a pre-configuration step in which a DiagnosticFimAliasEvent has been used. However, since the DiagnosticEvent in the role actualEvent exists, the Diagnostic Extract is considered complete with respect to this aspect of the Fim configuration.

⌋

[TPS_DEXT_01101] Consequence of the existence of DiagnosticFimAliasEventGroupMapping ⌈The references from meta-class DiagnosticFimAliasEventGroupMapping in the roles actualEvent and aliasEvent do not necessarily have to exist at the same time. The following rules apply:

Only actualEvent exists: In this case there is no DiagnosticFimAliasEventGroup available and the actual DiagnosticFimEventGroup can be taken directly for creating the Fim configuration.

Only aliasEvent exists: In this case the configuration is considered incomplete and only defines the intended semantics from the point of view of a pre-configuration of the Fim.

Both actualEvent and aliasEvent exist: In this case it is obvious that the configuration has undergone a pre-configuration step in which a DiagnosticFimAliasEventGroup has been used. However, since the DiagnosticFimEventGroup in the role actualEvent exists, the Diagnostic Extract is considered complete with respect to this aspect of the Fim configuration.

⌋

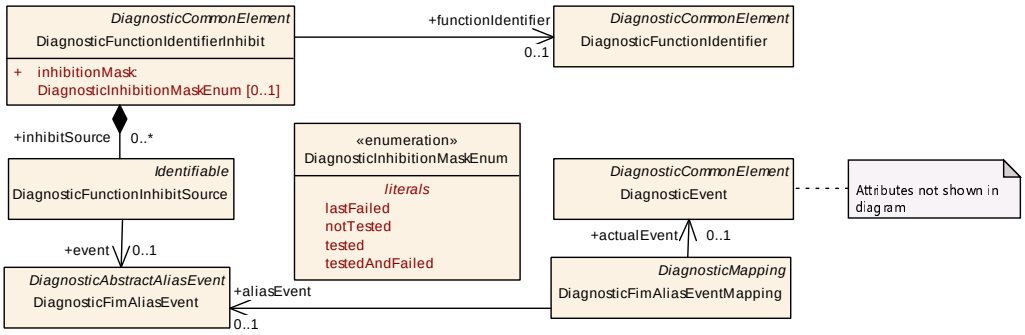

Figure 5.25: Inhibition of events in Fim for diagnostics configuration

Class	**DiagnosticFimAliasEventMapping**			
Note	This meta-class represents the ability to model the mapping of a DiagnosticEvent to a DiagnosticAlias Event. By this means the "preliminary" modeling by way of a DiagnosticAliasEvent is further substantiated. **Tags:** atp.recommendedPackage=DiagnosticFimEventMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
actualEvent	DiagnosticEvent	0..1	ref	This represents the reference to the actual diagnostic event. **Tags:** atp.Splitkey=actualEvent
aliasEvent	DiagnosticFimAlias Event	0..1	ref	This represents the reference to the alias event. **Tags:** atp.Splitkey=aliasEvent

Table 5.37: DiagnosticFimAliasEventMapping

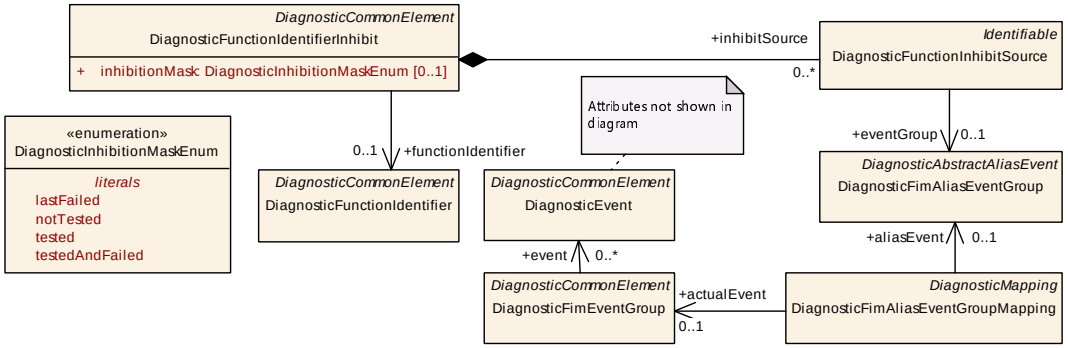

Figure 5.26: Inhibition of event groups in Fim for diagnostics configuration

Class	**DiagnosticFimAliasEventGroup**
Note	This meta-class represents the ability to define an alias for a Fim summarized event. This alias can be used in early phases of the configuration process until a further refinement is possible. **Tags:** atp.recommendedPackage=DiagnosticFimAliasEventGroups
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAbstractAliasEvent*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

△

Class	**DiagnosticFimAliasEventGroup**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
groupedAlias Event	DiagnosticFimAlias Event	*	ref	By means of this reference the grouping of Diagnostic AliasEvents within the DiagnosticFimSummaryEvent can be specified. **Tags:** atp.Splitkey=groupedAliasEvent

Table 5.38: DiagnosticFimAliasEventGroup

Class	**DiagnosticFimAliasEventGroupMapping**			
Note	This meta-class represents the ability to map a DiagnosticFimEventGroup to a DiagnosticFimAliasEvent Group. By this means the "preliminary" modeling by way of a DiagnosticFimAliasEventGroup is further substantiated. **Tags:** atp.recommendedPackage=DiagnosticFimAliasEventGroupMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
actualEvent	DiagnosticFimEvent Group	0..1	ref	This represents the reference to the actual summary event. **Tags:** atp.Splitkey=actualEvent
aliasEvent	DiagnosticFimAlias EventGroup	0..1	ref	This represents the reference to the alias summary event. **Tags:** atp.Splitkey=aliasEvent

Table 5.39: DiagnosticFimAliasEventGroupMapping

5.4.3 Mapping of Function Identifiers to Interested Software Components

Besides the ability to model a function inhibition itself, it is equally important that this inhibition can be queried from or pushed to interested software components.

This relation can be expressed by means of the `DiagnosticFimFunctionMapping`. The details are sketched in Figure 5.27.

[TPS_DEXT_01102] Semantics of `DiagnosticFimFunctionMapping` ⌈The meta-class `DiagnosticFimFunctionMapping` represents the ability to map a `DiagnosticFunctionIdentifier` to a `SwcServiceDependency` that addresses function inhibition.⌋

[constr_1454] `DiagnosticFimFunctionMapping` shall only reference a `SwcServiceDependency` that aggregates `FunctionInhibitionNeeds`

Imposition time: CP: `IT_DiagDes`

⌈A `DiagnosticFimFunctionMapping` shall only reference a `SwcServiceDependency` that aggregates `FunctionInhibitionNeeds` in the role `serviceNeeds`.⌋

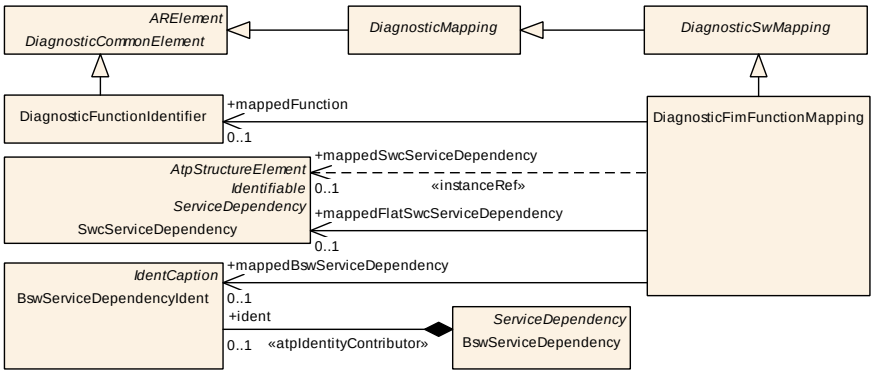

Figure 5.27: Mapping of function identifiers to the corresponding monitors

The modeling of this mapping is closely related to the existing modeling of mappings that involve `SwcServiceDependency` in the context of the diagnostic extract.

Class	**DiagnosticFimFunctionMapping**			
Note	This meta-class represents the ability to define a mapping between a function identifier (FID) and the corresponding SwcServiceDependency in the application software resp. basic software. **Tags:** atp.recommendedPackage=DiagnosticFimFunctionMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
mappedBsw Service Dependency	BswService DependencyIdent	0..1	ref	This is supposed to represent a reference to a Bsw ServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference. **Tags:** atp.Splitkey=mappedBswServiceDependency
mappedFlatSwc Service Dependency	SwcService Dependency	0..1	ref	This represents the ability to refer to an AtomicSw ComponentType that is available without the definition of how it will be embedded into the component hierarchy. **Tags:** atp.Splitkey=mappedFlatSwcServiceDependency
mapped Function	DiagnosticFunction Identifier	0..1	ref	This represents the mapped FID. **Tags:** atp.Splitkey=mappedFunction
mappedSwc Service Dependency	SwcService Dependency	0..1	iref	This represents the ability to point into the component hierarchy (under possible consideration of the root SoftwareComposition). **Tags:** atp.Splitkey=mappedSwcServiceDependency, mappedSwcServiceDependency.base, mappedSwc ServiceDependency.contextRootSwComposition, mappedSwcServiceDependency.contextSwComponent Prototype, mappedSwcServiceDependency.targetSwc ServiceDependency **InstanceRef implemented by:** SwcServiceDependency InSystemInstanceRef

Table 5.40: DiagnosticFimFunctionMapping

Class	FunctionInhibitionNeeds			
Note	Specifies the abstract needs on the configuration of the Function Inhibition Manager for one Function Identifier (FID). This class currently contains no attributes. Its name can be regarded as a symbol identifying the FID from the viewpoint of the component or module which owns this class.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 5.41: FunctionInhibitionNeeds

5.4.4 Diagnostic IUMPR to Function Identifier Mapping

[TPS_DEXT_01157] Semantics of `DiagnosticIumprToFunctionIdentifierMapping` ⌈The existence of a `DiagnosticIumprToFunctionIdentifierMapping` indicates that the computation of the `IUMPR` can be inhibited by the Fim.⌋

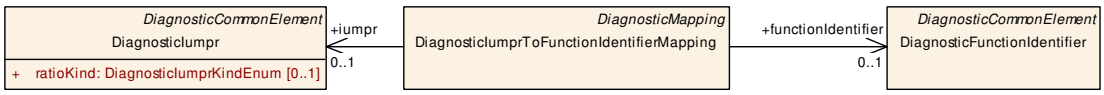

Figure 5.28: Modeling of the `DiagnosticIumprToFunctionIdentifierMapping`

Class	DiagnosticIumprToFunctionIdentifierMapping			
Note	This meta-class represents the ability to associate a DiagnosticFunctionIdentifier with a DiagnosticIumpr. **Tags:** atp.recommendedPackage=DiagnosticMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
function Identifier	DiagnosticFunction Identifier	0..1	ref	This reference identifies the applicable Diagnostic FunctionIdentifier. **Tags:** atp.Splitkey=functionIdentifier
iumpr	DiagnosticIumpr	0..1	ref	This reference identifies the applicable DiagnosticIumpr. **Tags:** atp.Splitkey=iumpr

Table 5.42: DiagnosticIumprToFunctionIdentifierMapping

[constr_10084] Existence of role `DiagnosticIumprToFunctionIdentifierMapping.iumpr`

Imposition time: CP: `IT_DiagDes`

⌈For all `DiagnosticIumprToFunctionIdentifierMapping`, the reference in the role `iumpr` shall exist.⌋

[constr_10085] Existence of role `DiagnosticIumprToFunctionIdentifierMapping.functionIdentifier`

Imposition time: CP: `IT_DiagDes`

⌈For all `DiagnosticIumprToFunctionIdentifierMapping`, the reference in the role `functionIdentifier` shall exist.⌋

5.5 J1939-related DiagnosticMapping

The relevant mappings in the context of diagnostics on J1939 are depicted in Figure 5.29.

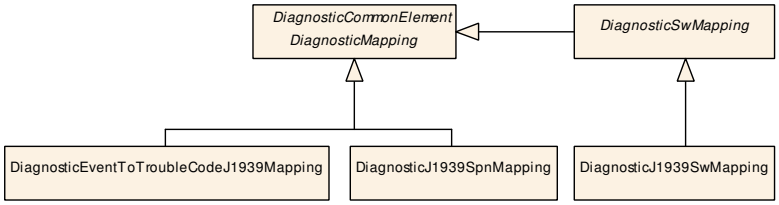

Figure 5.29: Modeling of the `DiagnosticMapping` in the context of J1939

The description of the modeling of the mapping in the context of diagnostics on J1939 can be roughly divided into the following sections:

- The `SPN` mapping is described in Chapter 5.5.1.
- The mapping between software components and controller applications is described in Chapter 5.5.2.
- The mapping between a diagnostic event and a J1939 `DTC` is described in Chapter 5.5.3.

5.5.1 SPN Mapping

[TPS_DEXT_01103] Semantics of meta-class `DiagnosticJ1939SpnMapping`

⌈The `DiagnosticJ1939SpnMapping` has the ability to associate a `DiagnosticJ1939Spn` with a `SystemSignal`. By this means it is possible to express that a given `SystemSignal` is taken to transport a J1939 *Suspect Parameter Number*.⌋

The modeling of the `DiagnosticJ1939SpnMapping` also implies that the same `SPN` can be sent by different `J1939NmNode`s. This ability is explicitly supported.

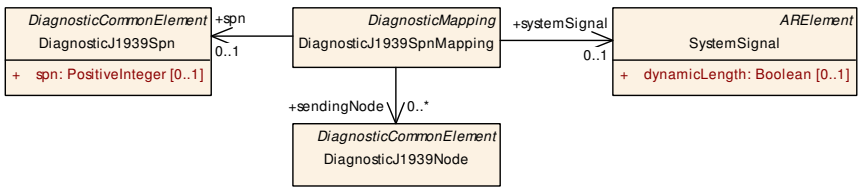

Figure 5.30: Modeling of the `DiagnosticJ1939SpnMapping`

[TPS_DEXT_01106] Relation of Controller Application to SPN ⌈It is technically possible that several *Controller Applications*, formally represented by the meta-class `DiagnosticJ1939Node`, can send the same *Suspect Parameter Number*.

In response to this specific characteristic of the J1939 approach, the multiplicity of the reference `DiagnosticJ1939SpnMapping.sendingNode` has been set to 0..*.⌋

Class	**DiagnosticJ1939SpnMapping**			
Note	This meta-class represents the ability to define a mapping between an SPN and a SystemSignal. The existence of a mapping means that neither the SPN nor the SystemSignal need to be updated if the relation between the two changes. **Tags:** atp.recommendedPackage=DiagnosticJ1939SpnMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
sendingNode	DiagnosticJ1939Node	*	ref	This additional reference has a supporting role in that it identifies all sending nodes of a given SPN. It is positively possible that a given SPN is sent by more than one node. Even tough the reference targets the DiagnosticJ1939 Node the semantics of the reference is bound to the J1939NmNode that is in turn referenced by the DiagnosticJ1939Node. **Tags:** atp.Splitkey=sendingNode
spn	DiagnosticJ1939Spn	0..1	ref	This reference goes to the SPN that shall be associated with a SystemSignal. **Tags:** atp.Splitkey=spn
systemSignal	SystemSignal	0..1	ref	This reference goes to the SystemSignal that shall be associated with an SPN. **Tags:** atp.Splitkey=systemSignal

Table 5.43: DiagnosticJ1939SpnMapping

Class	**DiagnosticJ1939Node**			
Note	This meta-class represents the diagnostic configuration of a J1939 Nm node, which in turn represents a "virtual Ecu" on the J1939 communication bus. **Tags:** atp.recommendedPackage=DiagnosticJ1939Nodes This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
nmNode	J1939NmNode	0..1	ref	This represents the reference to the "virtual Ecu" to which the enclosing DiagnosticJ1939Node is associated. **Tags:** atp.Splitkey=nmNode

Table 5.44: DiagnosticJ1939Node

5.5.2 Mapping between Software Components and Controller Applications

Another aspect of the modeling of J1939 diagnostics is that a mapping between the DiagnosticJ1939Node (which formally represents a *Controller Application*) and the AUTOSAR way of specifying a "function" (by means of the meta-class SwComponentPrototype) is required.

This leads to the definition of the DiagnosticJ1939SwMapping.

[TPS_DEXT_01108] Purpose of the DiagnosticJ1939SwMapping ⌈The purpose of the DiagnosticJ1939SwMapping is to associate a SwComponentPrototype with a DiagnosticJ1939Node. By this means a concrete functionality is mapped to the abstract J1939 *Controller Application*.⌋

The basis for this mapping is the existence of a CompositionSwComponentType rather than a System. The mapping can therefore (and this is the main motivation for this kind of modeling) be done independently of the deployment to a concrete project.

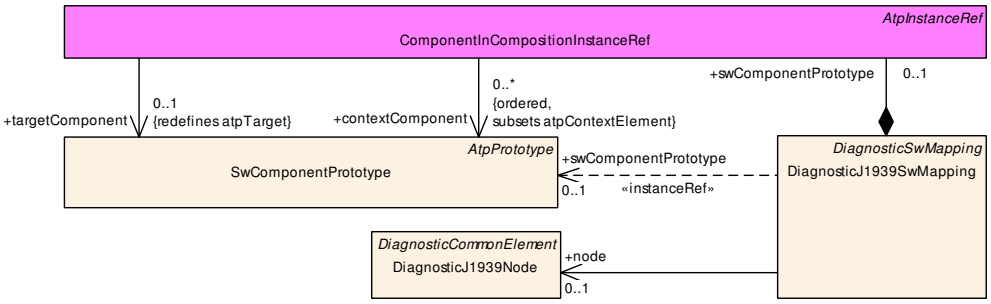

Figure 5.31: Mapping between Software Components and Controller Applications

Class	**DiagnosticJ1939SwMapping**			
Note	This meta-class represents the ability to map a piece of application software to a J1939DiagnosticNode. By this means the diagnostic configuration can be associated with the application software. **Tags:** atp.recommendedPackage=DiagnosticJ1939SwMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
node	DiagnosticJ1939Node	0..1	ref	This represents the mapped DiagnosticJ1939Node. **Tags:** atp.Splitkey=node
swComponent Prototype	SwComponent Prototype	0..1	iref	This represents the mapped SwComponentPrototype. **Tags:** atp.Splitkey=swComponentPrototype, sw ComponentPrototype.contextComponent, swComponent Prototype.targetComponent **InstanceRef implemented by:** ComponentIn CompositionInstanceRef

Table 5.45: DiagnosticJ1939SwMapping

5.5.3 Mapping between DiagnosticEvent and J1939 DTC

[TPS_DEXT_01145] Semantics of meta-class DiagnosticEventToTroubleCodeJ1939Mapping ⌈The DiagnosticEventToTroubleCodeJ1939Mapping is used to assign one (1:1) or multiple (n:1) DiagnosticEvents to a DiagnosticTroubleCodeJ1939.

In the case of an n:1 mapping, multiple instances of DiagnosticEventToTroubleCodeJ1939Mapping with the same reference in the role troubleCodeJ1939 but different references in the role diagnosticEvent have to be defined.⌋

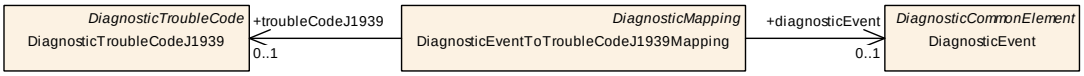

Figure 5.32: Mapping between DiagnosticEvent and DiagnosticTroubleCodeJ1939

Class	**DiagnosticEventToTroubleCodeJ1939Mapping**			
Note	By means of this meta-class it is possible to associate a DiagnosticEvent to a DiagnosticTroubleCode J1939. **Tags:** atp.recommendedPackage=DiagnosticMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to a DiagnosticEvent to which a J1939 Diagnostic Trouble Code is assigned. **Tags:** atp.Splitkey=diagnosticEvent
troubleCode J1939	DiagnosticTroubleCode J1939	0..1	ref	Reference to a J1939 Diagnostic Trouble Code to which a DiagnosticEvent is assigned. **Tags:** atp.Splitkey=troubleCodeJ1939

Table 5.46: DiagnosticEventToTroubleCodeJ1939Mapping

[constr_1857] Existence of the reference DiagnosticEventToTroubleCodeJ1939Mapping.diagnosticEvent

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEventToTroubleCodeJ1939Mapping, the reference in the role diagnosticEvent shall exist.⌋

[constr_1858] Existence of the attribute DiagnosticEventToTroubleCodeJ1939Mapping.troubleCodeJ1939

Imposition time: CP: IT_DiagDes

⌈For each DiagnosticEventToTroubleCodeJ1939Mapping, attribute troubleCodeJ1939 shall exist.⌋

5.6 Diagnostic Configuration of CP Software Clusters

On the *AUTOSAR classic platform*, the so-called “CP Software Cluster” (formalized by meta-class `CpSoftwareCluster`) provides an approach for a modular decomposition of an `ECU` into components that can be independently developed and tested.

Thanks to its modular and decentralized nature, the Diagnostic Extract is capable of supporting the diagnostic configuration of a CP Software Cluster.

However, there are some aspects in which the development and integration of CP Software Clusters with the corresponding Diagnostic Extract can be improved. These aspects are described in this chapter.

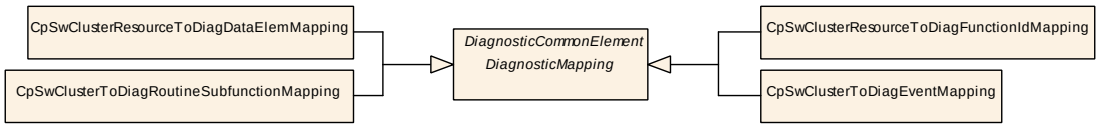

Figure 5.33: Subclasses of `DiagnosticMapping` that are relevant for the configuration of `CpSoftwareCluster`

The description of the modeling of the mapping in the context of CP software clusters can be roughly divided into the following sections:

- The association of the diagnostic mapping with the CP software cluster is described in Chapter 5.6.1.
- The mapping between a DEXT elements and software cluster resources is described in Chapter 5.6.2.
- The mapping between a diagnostic event and software cluster resources is described in Chapter 5.6.3.
- The mapping between a diagnostic data element and software cluster resources is described in Chapter 5.6.4.
- The mapping between a diagnostic routines and software cluster resources is described in Chapter 5.6.5.
- The mapping between a function identifiers and software cluster resources is described in Chapter 5.6.6.

5.6.1 Association of Diagnostic Mapping with a CP Software Cluster

[TPS_DEXT_01160] Semantics of the reference in the role `DiagnosticMapping.providerSoftwareCluster` ⌈In early stages of the design of a CP Software Cluster, it is possible to express the relevance of a given `DiagnosticMapping` for a specific `CpSoftwareCluster` that is not yet detailed enough for the `DiagnosticMapping` to refer to the final target of the mapping on the side of the application software.

For this purpose, the reference in the role DiagnosticMapping.providerSoftwareCluster can be used to refer to a CP Software Cluster in a *providing* role.

Conversely, the reference in the role DiagnosticMapping.requesterSoftwareCluster can be used to refer to a CP Software Cluster in a *requesting* role.⌋

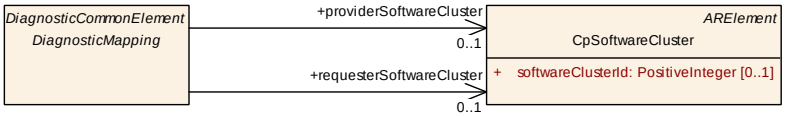

Figure 5.34: Reference from DiagnosticMapping to CpSoftwareCluster

5.6.2 Mapping between DEXT elements and CP Software Cluster Resources

Meta-class CpSoftwareClusterResource is described in the [5, TPS System Template]. It represents a location in memory that is reserved e.g. for conveying information between CP Software Clusters.

By creating a mapping between an element of a Diagnostic Extract and a CpSoftwareClusterResource it is possible to already designate specific CpSoftwareClusterResources for the communication of diagnostic information even before the SwComponentTypes and SwComponentPrototypes within the CpSoftwareCluster are fully specified, i.e. the creation of a complete DiagnosticMapping becomes possible.

Class	***CpSoftwareClusterResource*** (abstract)			
Note	Represents a single resource required or provided by a CP Software Cluster. **Tags:** atp.recommendedPackage=Resources This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CpSoftwareClusterCommunicationResource, CpSoftwareClusterServiceResource			
Aggregated by	CpSoftwareClusterResourcePool.resource			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dependentResource	RoleBasedResourceDependency	*	aggr	Link to a resource which depends on this resource to implement them. **Tags:** atp.Splitkey=dependentResource
globalResourceId	PositiveInteger	0..1	attr	A unique identifiers per resource used for the connection process. The identifier is required to be unique in the scope of a single machine. If software clusters are designed to be reused on multiple machines the uniqueness requirements applies for all the intended machines.
isMandatory	Boolean	0..1	attr	This attribute indicates, that the resource is mandatory to operate the Software Cluster. If the resource is not provided on the machine the connection process of any Software Cluster requiring this resource gets aborted.

Table 5.47: CpSoftwareClusterResource

Class	RoleBasedResourceDependency			
Note	This class specifies a dependency between CpSoftwareClusterResources. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	*CpSoftwareClusterResource*.dependentResource			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
resource	CpSoftwareCluster Resource	0..1	ref	Reference to resource for which the dependency is depicted. **Tags:** atp.Splitkey=resource
role	Identifier	0..1	attr	This is attributes characterizes the kind of dependency

Table 5.48: RoleBasedResourceDependency

5.6.3 Mapping between Diagnostic Events and CP Software Cluster Resources

[TPS_DEXT_01161] Mapping between a `DiagnosticEvent` and a `CpSoftwareClusterResource` ⌈The creation of a mapping between a `DiagnosticEvent` and a `CpSoftwareClusterResource` by means of a `CpSwClusterToDiagEventMapping` supports the configuration of the reporting of diagnostic events or the querying of the event status across the boundaries of a `CpSoftwareCluster`.⌋

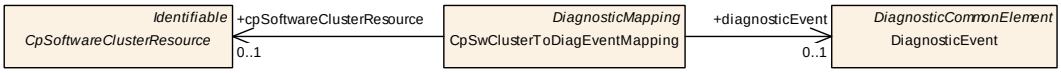

Figure 5.35: Mapping between `DiagnosticEvent` and `CpSoftwareClusterResource`

Class	CpSwClusterToDiagEventMapping			
Note	This meta-class represents the ability to associate a CpSoftwareClusterResource with a Diagnostic Event. This allows for indicating that the CpSoftwareClusterResource is used to convey the reporting or status query of the mapped DiagnosticEvent. **Tags:** atp.recommendedPackage=CpSoftwareClusterToDiagMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
cpSoftware Cluster Resource	CpSoftwareCluster Resource	0..1	ref	This reference identifies the mapped CpSoftwareCluster Resource. **Tags:** atp.Splitkey=cpSoftwareClusterResource
diagnosticEvent	DiagnosticEvent	0..1	ref	This reference identifies the mapped DiagnosticEvent. **Tags:** atp.Splitkey=diagnosticEvent

Table 5.49: CpSwClusterToDiagEventMapping

5.6.4 Mapping between Diagnostic Data Elements and CP Software Cluster Resources

[TPS_DEXT_01162] Creation of a mapping between a DiagnosticDataElement and a CpSoftwareClusterResource ⌈The creation of a mapping between a DiagnosticDataElement and a CpSoftwareClusterResource by means of a CpSwClusterResourceToDiagDataElemMapping supports the configuration of the transport of an element of a DID across the boundaries of a CpSoftwareCluster.⌋

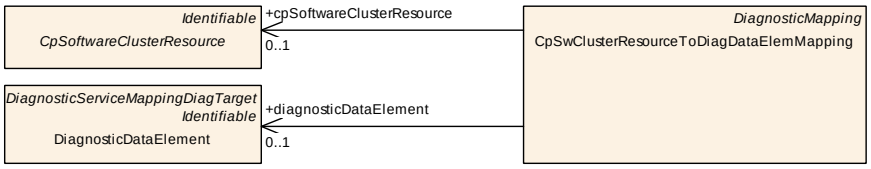

Figure 5.36: Mapping between DiagnosticDataElement and CpSoftwareClusterResource

Class	**CpSwClusterResourceToDiagDataElemMapping**			
Note	This meta-class represents the ability to associate a CpSoftwareClusterResource with a DiagnosticData Element. This allows for indicating that the CpSoftwareClusterResource is used to convey the Diagnostic DataElement. **Tags:** atp.recommendedPackage=CpSoftwareClusterToDiagMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
cpSoftware Cluster Resource	CpSoftwareCluster Resource	0..1	ref	This represents the affected CpSoftwareClusterResource. **Tags:** atp.Splitkey=cpSoftwareClusterResource
diagnosticData Element	DiagnosticDataElement	0..1	ref	This reference represents the affected DiagnosticData Element. **Tags:** atp.Splitkey=diagnosticDataElement

Table 5.50: CpSwClusterResourceToDiagDataElemMapping

5.6.5 Mapping between Diagnostic Routines and CP Software Cluster Resources

[TPS_DEXT_01163] Mapping between a DiagnosticRoutineSubfunction and a CpSoftwareClusterResource ⌈The creation of a mapping between a concrete sub-class of DiagnosticRoutineSubfunction and a CpSoftwareClusterResource by means of a CpSwClusterToDiagRoutineSubfunctionMapping supports the configuration of the transport of call and return arguments of a diagnostic routine call across the boundaries of a CpSoftwareCluster.⌋

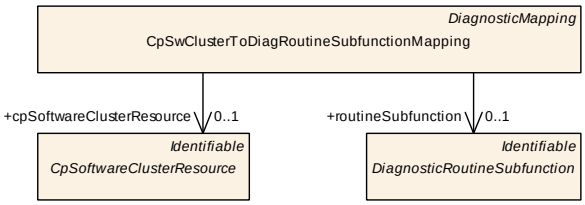

Figure 5.37: Mapping between DiagnosticRoutineSubfunction and CpSoftwareClusterResource

Class	**CpSwClusterToDiagRoutineSubfunctionMapping**			
Note	This meta-class represents the ability to associate a CpSoftwareClusterResource with a subfunction of a DiagnosticRoutine. This allows for indicating that the CpSoftwareClusterResource is used to convey the calling or result return of the mapped DiagnosticRoutine. **Tags:** atp.recommendedPackage=CpSoftwareClusterToDiagMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
cpSoftware Cluster Resource	CpSoftwareCluster Resource	0..1	ref	This reference identifies the mapped CpSoftwareCluster Resource. **Tags:** atp.Splitkey=cpSoftwareClusterResource
routine Subfunction	DiagnosticRoutine Subfunction	0..1	ref	This reference identifies the mapped subfunction of a DiagnosticRoutine. **Tags:** atp.Splitkey=routineSubfunction

Table 5.51: CpSwClusterToDiagRoutineSubfunctionMapping

5.6.6 Mapping between Diagnostic Function Identifiers and CP Software Cluster Resources

[TPS_DEXT_01164] Mapping between a DiagnosticFunctionIdentifier and a CpSoftwareClusterResource ⌈The creation of a mapping between a DiagnosticFunctionIdentifier and a CpSoftwareClusterResource by means of a CpSwClusterResourceToDiagFunctionIdMapping supports the configuration of the transport of the execution permission associated with a DiagnosticFunctionIdentifier across the boundaries of a CpSoftwareCluster.⌋

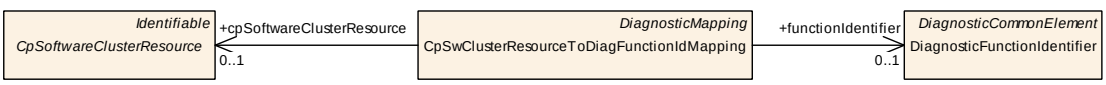

Figure 5.38: Mapping between DiagnosticFunctionIdentifier and CpSoftwareClusterResource

Class	**CpSwClusterResourceToDiagFunctionIdMapping**			
Note	This meta-class represents the ability to associate a CpSoftwareClusterResource with a subfunction of a DiagnosticFunctionIdentifier. This allows for indicating that the CpSoftwareClusterResource is used to convey the execution permission associated with the mapped function identifier. **Tags:** atp.recommendedPackage=CpSoftwareClusterToDiagMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
cpSoftware Cluster Resource	CpSoftwareCluster Resource	0..1	ref	This reference identifies the mapped CpSoftwareCluster Resource. **Tags:** atp.Splitkey=cpSoftwareClusterResource
function Identifier	DiagnosticFunction Identifier	0..1	ref	This reference identifies the mapped DiagnosticFunction Identifier. **Tags:** atp.Splitkey=functionIdentifier

Table 5.52: CpSwClusterResourceToDiagFunctionIdMapping

A Reference Material

This chapter contains some relevant reference material for this specification.

A.1 Imposition Times of Constraints

The constraints formulated in this document could have different *actual* imposition times which denote the steps in the workflow when the respective constraint has to be imposed.

Some imposition times "include" other imposition times, an example for this relation is discussed in the Table A.1

The imposition times that are considered applicable in the scope of this document[1] are listed in Table A.1.

Please note that the imposition times are intentionally rendered as technical terms such that it is possible to link back from each constraint to the definition of the affected imposition time in Table A.1.

Imposition Time	Description	Motivation
IT_DiagDes	Diagnostic design is completed	This imposition time denotes the step in the workflow, where the diagnostic design is about to be finished independent of the standard where it is applied. This includes the finalization, e.g., of the DEXT.

Table A.1: Imposition Times of constraints in this document

The constraints in this document are also applicable to the *AUTOSAR adaptive platform*, unless a constraint exist in the context of the [7, TPS Manifest Specification] that overrules a corresponding constraint in this document.

A.2 Abbreviations

The content of this appendix chapter is *informative* in nature and shall **not** be considered as *normative* content.

The following table contains a list of abbreviations used in the scope of this document along with the spelled-out meaning of each of the abbreviations.

Abbreviation	*meaning*
API	Application Programming Interface
ARXML	AUTOSAR XML
BSW	Basic Software

▽

[1] Different imposition times may be defined in the context of other AUTOSAR standard documents

△

Abbreviation	*meaning*
BswM	Basic Software Manager
CAN	Controller Area Network
CSE	Codes for Scaling Units
DCM	Diagnostics Communication Manager
DCY	Driving Cycle
DEM	Diagnostics Event Manager
DID	Diagnostic Identifier
DTC	Diagnostic Trouble Code
DTR	Diagnostic Test Result
DoIP	Diagnostics over IP
ECU	Electrical Control Unit
EcuC	ECU Configuration
FID	Function Identifier
FIM	Function Inhibition Manager
GID	Group Identifier
ID	Identifier
IO	Input/Output
IP	Internet Protocol
IUMPR	In-Use Monitor Performance Ratio
ISO	International Organization for Standardization
LIN	Local Interconnect Network
NRC	Negative Response Code
OBD	On-Board Diagnostic
ODX	Open Diagnostic Data Exchange
OEM	Original Equipment Manufacturer
PDU	Protocol Data Unit
PID	Parameter Identifier
PKI	Public-Key Infrastructure
PTO	Power Take Off
RA	Routing Activation
RAM	Random Access Memory
RID	Routine Identifier
ROE	Response on Event
ROM	Read-Only Memory
RTE	Run-TIme Environment
RS	Requirements Specification
RX	Receive
SPN	Suspect Parameter Number
SW	Software
SWC	Software Component
SWCD	Software Component Description
TID	Test Identifier
TPS	Template Specification

▽

△

Abbreviation	*meaning*
TX	Transmit
SWS	Software Specification
UDS	Unified Diagnostic Services
UML	Unified Modeling Language
VFB	Virtual Functional Bus
VIN	Vehicle Identification Number
WWH-OBD	World-Wide Harmonized On-Board Diagnostics
XML	Extensible Markup Language
XSD	XML Schema Definition

Table A.2: Abbreviations used in the scope of this Document

B Examples

The content of this appendix chapter is *informative* in nature and shall **not** be considered as *normative* content.

This section contains a collection of examples for **select modeling aspects** discussed in this document.

Please note that it is not foreseen to create examples for every single aspect of using the model elements discussed in this document.

B.1 Examples of the Definition of nested Diagnostic Parameters

This section contains examples for the definition of nested DiagnosticParameters. The background is explained in Chapter 4.2.3.

B.1.1 Simple Nesting Example

The most simple example for a nested DiagnosticParameter is sketched in Figure B.1. It consists of just one nesting level and the aggregated DiagnosticParameterElements named “X” and “Y” that are located – one after the other – at the start of the enclosing DiagnosticParameter named “A”.

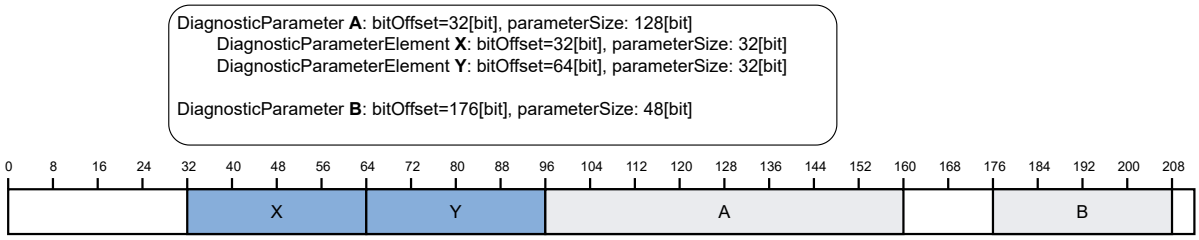

Figure B.1: Example for the definition of a simple structure inside the definition of a DiagnosticParameter

As sketched in Figure B.1, the nested DiagnosticParameterElements **do not cover the entire slot** reserved by the enclosing DiagnosticParameter.

The free space that is not allocated for the nested DiagnosticParameterElements **cannot be claimed by other** DiagnosticParameters, it is reserved for future extensions of the enclosing DiagnosticParameter “A”.

The simplified (i.e. stripped down to the bare minimum) modeling of the DiagnosticDataIdentifier (which would look similar for the modeling of a DiagnosticRoutineSubfunction) sketched in Figure B.1 is contained in Listing B.1.

```
<DIAGNOSTIC-DATA-IDENTIFIER>
  <SHORT-NAME>NestedAligned</SHORT-NAME>
  <DATA-ELEMENTS>
    <DIAGNOSTIC-PARAMETER>
      <BIT-OFFSET>32</BIT-OFFSET>
```

```
      <PARAMETER-SIZE>128</PARAMETER-SIZE>
     <IDENT>
        <SHORT-NAME>A</SHORT-NAME>
        <SUB-ELEMENTS>
          <DIAGNOSTIC-PARAMETER-ELEMENT>
            <SHORT-NAME>X</SHORT-NAME>
            <CATEGORY>LEAF</CATEGORY>
            <BIT-OFFSET>32</BIT-OFFSET>
            <DATA-ELEMENTS>
              <DIAGNOSTIC-DATA-ELEMENT>
                <SHORT-NAME>X</SHORT-NAME>
              </DIAGNOSTIC-DATA-ELEMENT>
            </DATA-ELEMENTS>
          </DIAGNOSTIC-PARAMETER-ELEMENT>
          <DIAGNOSTIC-PARAMETER-ELEMENT>
            <SHORT-NAME>Y</SHORT-NAME>
            <CATEGORY>LEAF</CATEGORY>
            <BIT-OFFSET>64</BIT-OFFSET>
            <DATA-ELEMENTS>
              <DIAGNOSTIC-DATA-ELEMENT>
                <SHORT-NAME>Y</SHORT-NAME>
              </DIAGNOSTIC-DATA-ELEMENT>
            </DATA-ELEMENTS>
          </DIAGNOSTIC-PARAMETER-ELEMENT>
        </SUB-ELEMENTS>
      </IDENT>
    </DIAGNOSTIC-PARAMETER>
  </DATA-ELEMENTS>
</DIAGNOSTIC-DATA-IDENTIFIER>
```

Listing B.1: Example for the definition of a simple structure with aligned `DiagnosticParameterElement` inside the definition of a `DiagnosticParameter`

B.1.2 Simple Nesting Example with Gaps

Another, slightly more complicated example is sketched in Figure B.2. In contrast to the example depicted in Figure B.1, the two sketched `DiagnosticParameterElement`s named “X” and “Y” are not aligned with the start location of the enclosing `DiagnosticParameter` and each other.

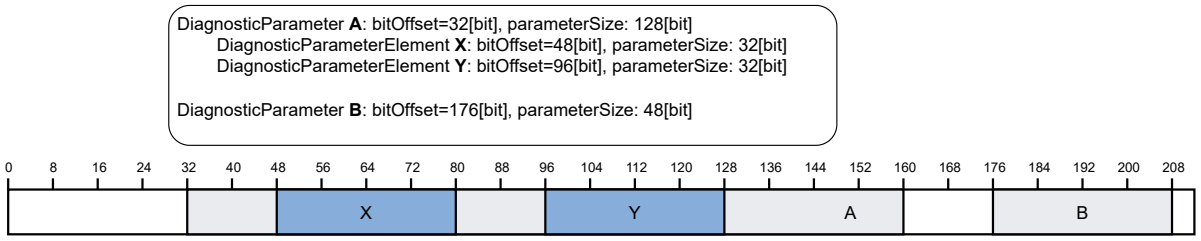

Figure B.2: Example for the definition of a simple structure with gaps inside the definition of a `DiagnosticParameter`

In this specific case, future extensions of the enclosing `DiagnosticParameter` can be done in the free space at the start, in between “X” and “Y”, and up to the tail of the slot claimed by the enclosing `DiagnosticParameter` “A”.

The simplified (i.e. stripped down to the bare minimum) modeling of the `DiagnosticDataIdentifier` (which would look similar for the modeling of a `DiagnosticRoutineSubfunction`) sketched in Figure B.2 is contained in Listing B.2.

```
<DIAGNOSTIC-DATA-IDENTIFIER>
  <SHORT-NAME>NestedNotAligned</SHORT-NAME>
  <DATA-ELEMENTS>
    <DIAGNOSTIC-PARAMETER>
      <BIT-OFFSET>32</BIT-OFFSET>
      <PARAMETER-SIZE>128</PARAMETER-SIZE>
      <IDENT>
        <SHORT-NAME>A</SHORT-NAME>
        <SUB-ELEMENTS>
          <DIAGNOSTIC-PARAMETER-ELEMENT>
            <SHORT-NAME>X</SHORT-NAME>
            <CATEGORY>LEAF</CATEGORY>
            <BIT-OFFSET>48</BIT-OFFSET>
            <DATA-ELEMENTS>
              <DIAGNOSTIC-DATA-ELEMENT>
                <SHORT-NAME>X</SHORT-NAME>
              </DIAGNOSTIC-DATA-ELEMENT>
            </DATA-ELEMENTS>
          </DIAGNOSTIC-PARAMETER-ELEMENT>
          <DIAGNOSTIC-PARAMETER-ELEMENT>
            <SHORT-NAME>Y</SHORT-NAME>
            <CATEGORY>LEAF</CATEGORY>
            <BIT-OFFSET>96</BIT-OFFSET>
            <DATA-ELEMENTS>
              <DIAGNOSTIC-DATA-ELEMENT>
                <SHORT-NAME>Y</SHORT-NAME>
              </DIAGNOSTIC-DATA-ELEMENT>
            </DATA-ELEMENTS>
          </DIAGNOSTIC-PARAMETER-ELEMENT>
        </SUB-ELEMENTS>
      </IDENT>
    </DIAGNOSTIC-PARAMETER>
  </DATA-ELEMENTS>
</DIAGNOSTIC-DATA-IDENTIFIER>
```

Listing B.2: Example for the definition of a simple structure with aligned `DiagnosticParameterElement` inside the definition of a `DiagnosticParameter`

B.1.3 Two-Level Nesting Example

As emphasized by the example in Figure B.3, it would be possible to specify a structure inside a `DiagnosticParameter` that has gaps

- at the start, i.e. between the start of the `DiagnosticParameter` and the first `DiagnosticParameterElement`,
- in between the placement of `DiagnosticParameterElement`s,
- at the tail, i.e. between the placement of the last `DiagnosticParameterElement` and the tail of the `DiagnosticParameter`.

In other words, it is not required that sub-elements are placed one directly after the other in their respective context.

Similarly, a `DID` that hosts a `DiagnosticParameter` could also have gaps.

- at the start, i.e.s between the start of the `DiagnosticAbstractDataIdentifier` resp. `DiagnosticRoutineSubfunction` and the first `DiagnosticParameter`,

- in between the placement of DiagnosticParameters,
- at the tail, i.e.s between the placement of the last DiagnosticParameter and the tail of the DiagnosticAbstractDataIdentifier/DiagnosticRoutineSubfunction.

As indicated by Figure 4.2, the nested definition of a DiagnosticParameter is applied by utilizing the roles

- DiagnosticParameter.ident.subElement and
- DiagnosticParameterElement.subElement.

An example for the definition of a nested structure inside the definition of DiagnosticParameter can be found in Figure B.3.

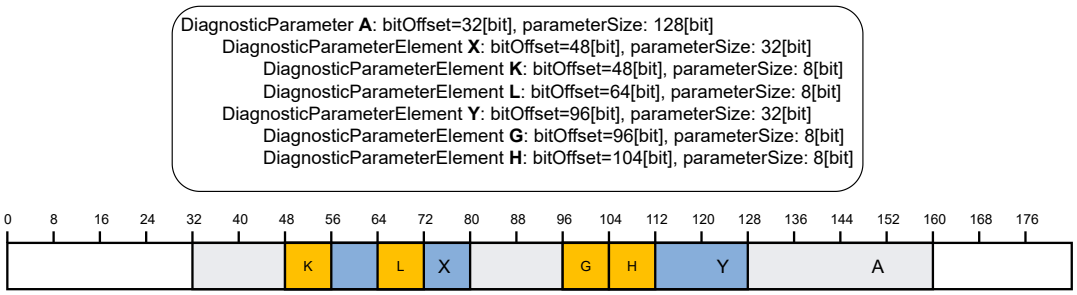

Figure B.3: Example for the definition of a nested structure inside the definition of a DiagnosticParameter

The example postulates the existence of a DiagnosticParameter named “A” inside the definition of a given DiagnosticDataIdentifier (which could also be a DiagnosticRoutineSubfunction).

The DiagnosticParameter named “A” defines two subElements named “X” and “Y”. “X”, in turn, defines two subElements named “K” and “L”. Conversely, “Y” has two subElements named “G” and “H”.

The serialized model content contained in Listing B.3 picks up the example depicted in Figure B.3 and describes how a very minimal modeling of the essential model elements looks like.

```
<DIAGNOSTIC-DATA-IDENTIFIER>
  <SHORT-NAME>NestedMapping</SHORT-NAME>
  <DATA-ELEMENTS>
    <DIAGNOSTIC-PARAMETER>
      <BIT-OFFSET>32</BIT-OFFSET>
      <PARAMETER-SIZE>128</PARAMETER-SIZE>
      <IDENT>
        <SHORT-NAME>A</SHORT-NAME>
        <SUB-ELEMENTS>
          <DIAGNOSTIC-PARAMETER-ELEMENT>
            <SHORT-NAME>X</SHORT-NAME>
            <CATEGORY>STRUCTURE</CATEGORY>
            <BIT-OFFSET>48</BIT-OFFSET>
            <PARAMETER-SIZE>32</PARAMETER-SIZE>
            <SUB-ELEMENTS>
              <DIAGNOSTIC-PARAMETER-ELEMENT>
                <SHORT-NAME>K</SHORT-NAME>
                <CATEGORY>LEAF</CATEGORY>
```

```
            <BIT-OFFSET>48</BIT-OFFSET>
            <DATA-ELEMENTS>
              <DIAGNOSTIC-DATA-ELEMENT>
                <SHORT-NAME>K</SHORT-NAME>
              </DIAGNOSTIC-DATA-ELEMENT>
            </DATA-ELEMENTS>
          </DIAGNOSTIC-PARAMETER-ELEMENT>
          <DIAGNOSTIC-PARAMETER-ELEMENT>
            <SHORT-NAME>L</SHORT-NAME>
            <CATEGORY>LEAF</CATEGORY>
            <BIT-OFFSET>64</BIT-OFFSET>
            <DATA-ELEMENTS>
              <DIAGNOSTIC-DATA-ELEMENT>
                <SHORT-NAME>L</SHORT-NAME>
              </DIAGNOSTIC-DATA-ELEMENT>
            </DATA-ELEMENTS>
          </DIAGNOSTIC-PARAMETER-ELEMENT>
        </SUB-ELEMENTS>
      </DIAGNOSTIC-PARAMETER-ELEMENT>
      <DIAGNOSTIC-PARAMETER-ELEMENT>
        <SHORT-NAME>Y</SHORT-NAME>
        <CATEGORY>STRUCTURE</CATEGORY>
        <BIT-OFFSET>96</BIT-OFFSET>
        <PARAMETER-SIZE>32</PARAMETER-SIZE>
        <SUB-ELEMENTS>
          <DIAGNOSTIC-PARAMETER-ELEMENT>
            <SHORT-NAME>G</SHORT-NAME>
            <CATEGORY>LEAF</CATEGORY>
            <BIT-OFFSET>96</BIT-OFFSET>
            <DATA-ELEMENTS>
              <DIAGNOSTIC-DATA-ELEMENT>
                <SHORT-NAME>G</SHORT-NAME>
              </DIAGNOSTIC-DATA-ELEMENT>
            </DATA-ELEMENTS>
          </DIAGNOSTIC-PARAMETER-ELEMENT>
          <DIAGNOSTIC-PARAMETER-ELEMENT>
            <SHORT-NAME>H</SHORT-NAME>
            <CATEGORY>LEAF</CATEGORY>
            <BIT-OFFSET>104</BIT-OFFSET>
            <DATA-ELEMENTS>
              <DIAGNOSTIC-DATA-ELEMENT>
                <SHORT-NAME>H</SHORT-NAME>
              </DIAGNOSTIC-DATA-ELEMENT>
            </DATA-ELEMENTS>
          </DIAGNOSTIC-PARAMETER-ELEMENT>
        </SUB-ELEMENTS>
      </DIAGNOSTIC-PARAMETER-ELEMENT>
    </SUB-ELEMENTS>
  </IDENT>
</DIAGNOSTIC-PARAMETER>
</DATA-ELEMENTS>
</DIAGNOSTIC-DATA-IDENTIFIER>
```

Listing B.3: Example for the definition of a nested structure inside the definition of a `DiagnosticParameter`

B.1.4 Array of Struct Example

This example (as depicted in Figure B.4) assumes that the `DiagnosticParameter` is representing an array of RGB color values that are stored in a structure of three elements (representing the R, G, and B values).

The example also shows that it is possible to define the size of the array element (in this case: 32 bit) greater that the actually used data real estate (in this case 24 bit) occupied by the mentioned structure in each array element.

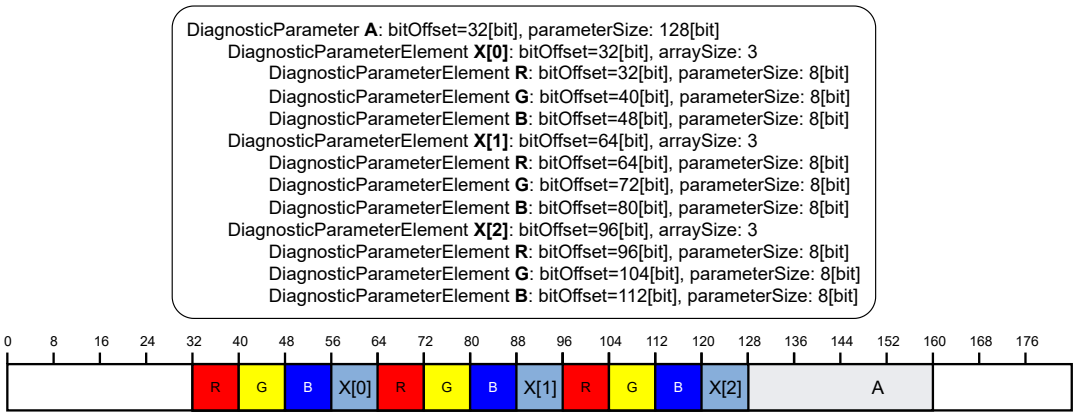

Figure B.4: Example for the definition of a `DiagnosticParameter` that represents an array of structures

The model depicted in Figure B.4 is also sketched in a minimal way in Listing B.4.

```
<DIAGNOSTIC-DATA-IDENTIFIER>
  <SHORT-NAME>DidArrayOfStruct</SHORT-NAME>
  <DATA-ELEMENTS>
    <DIAGNOSTIC-PARAMETER>
      <BIT-OFFSET>32</BIT-OFFSET>
      <PARAMETER-SIZE>128</PARAMETER-SIZE>
      <IDENT>
        <SHORT-NAME>A</SHORT-NAME>
        <SUB-ELEMENTS>
          <DIAGNOSTIC-PARAMETER-ELEMENT>
            <SHORT-NAME>X</SHORT-NAME>
            <CATEGORY>ARRAY</CATEGORY>
            <BIT-OFFSET>32</BIT-OFFSET>
            <PARAMETER-SIZE>32</PARAMETER-SIZE>
            <ARRAY-SIZE>3</ARRAY-SIZE>
            <SUB-ELEMENTS>
              <DIAGNOSTIC-PARAMETER-ELEMENT>
                <SHORT-NAME>R</SHORT-NAME>
                <CATEGORY>LEAF</CATEGORY>
                <BIT-OFFSET>32</BIT-OFFSET>
                <DATA-ELEMENTS>
                  <DIAGNOSTIC-DATA-ELEMENT>
                    <SHORT-NAME>R</SHORT-NAME>
                  </DIAGNOSTIC-DATA-ELEMENT>
                </DATA-ELEMENTS>
              </DIAGNOSTIC-PARAMETER-ELEMENT>
              <DIAGNOSTIC-PARAMETER-ELEMENT>
                <SHORT-NAME>G</SHORT-NAME>
                <CATEGORY>LEAF</CATEGORY>
                <BIT-OFFSET>40</BIT-OFFSET>
                <DATA-ELEMENTS>
                  <DIAGNOSTIC-DATA-ELEMENT>
                    <SHORT-NAME>G</SHORT-NAME>
                  </DIAGNOSTIC-DATA-ELEMENT>
                </DATA-ELEMENTS>
              </DIAGNOSTIC-PARAMETER-ELEMENT>
              <DIAGNOSTIC-PARAMETER-ELEMENT>
                <SHORT-NAME>B</SHORT-NAME>
                <CATEGORY>LEAF</CATEGORY>
                <BIT-OFFSET>48</BIT-OFFSET>
                <DATA-ELEMENTS>
                  <DIAGNOSTIC-DATA-ELEMENT>
```

```
                    <SHORT-NAME>B</SHORT-NAME>
                  </DIAGNOSTIC-DATA-ELEMENT>
                </DATA-ELEMENTS>
              </DIAGNOSTIC-PARAMETER-ELEMENT>
            </SUB-ELEMENTS>
          </DIAGNOSTIC-PARAMETER-ELEMENT>
        </SUB-ELEMENTS>
      </IDENT>
    </DIAGNOSTIC-PARAMETER>
  </DATA-ELEMENTS>
</DIAGNOSTIC-DATA-IDENTIFIER>
```

Listing B.4: Example for the definition of an array of structures inside the definition of a `DiagnosticParameter`

B.2 Examples of Diagnostic Service Data Mapping

This section provides examples of mapping scenarios that target more or less complex data objects defined by means of a `DiagnosticParameter` and `DiagnosticParameterElement` as well as the corresponding definition of a composite `AutosarDataPrototype`.

The background is explained in Chapter 5.2.1.

B.2.1 Simple Structure

Figure B.5 provides a sketch of a simple example situation for the application of [TPS_DEXT_01179].

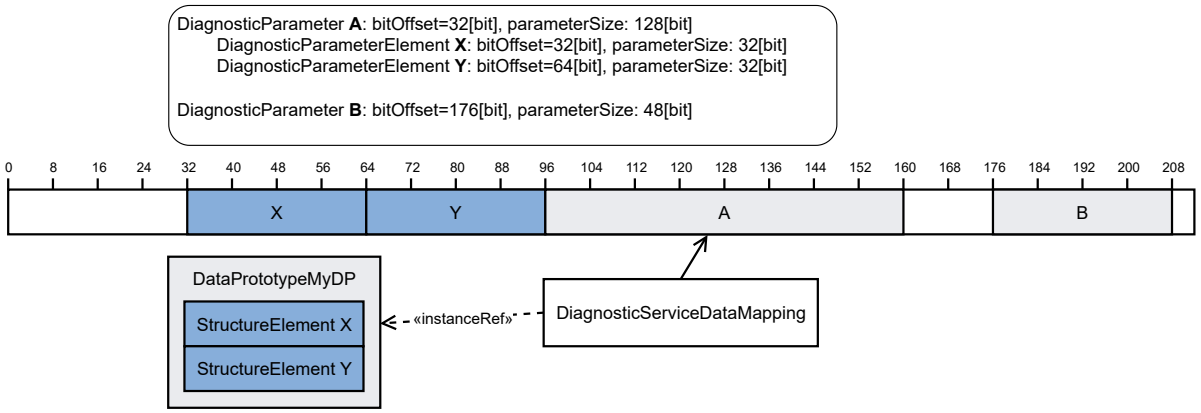

Figure B.5: Example for the mapping of a nested structure to an aligned `DiagnosticParameter`

As indicated in the example, the mapped `DiagnosticParameter` is larger than required and therefore leaves room for a later extension of the mapped data structure without the need to update the layout of the `DiagnosticDataIdentifier`.

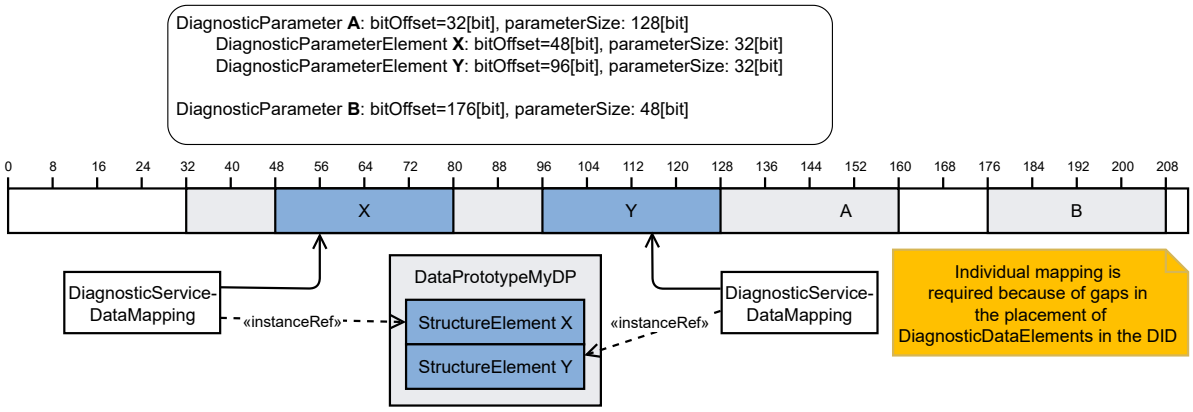

Figure B.6: Example for the mapping of a simple nested structure to a DiagnosticParameter

[TPS_DEXT_01180] Existence of reference to DiagnosticParameterElement in the role diagnosticParameter ⌈If the reference DiagnosticServiceDataMapping.diagnosticParameter exists, then the intention is to map a primitive (or composite) DataPrototype to a DiagnosticParameterElement in one piece.

This means that the elements of the composite DataPrototype would not have to be placed one after the other starting at the bitOffset of the referenced DiagnosticParameter. Gaps are positively supported.⌋

The consequence of [TPS_DEXT_01180] is sketched in Figure B.6 for the case of a simple nesting of non-aligned DiagnosticParameterElements.

B.2.2 Nested Structure

A more complex situation is sketched in Figure B.7. In this case, the mapping is configured such that “leaf” elements named “K” and “L” are mapped and there is also the depiction of how the mapping on an intermediate level (DiagnosticParameterElement named “Y”) would look like.

Please note that (as indicated in the sketch depicted in Figure B.7), individual mappings are required for situations where gaps exist in the placement of DiagnosticParameterElements in the context of the enclosing DiagnosticParameter or DiagnosticParameterElement.

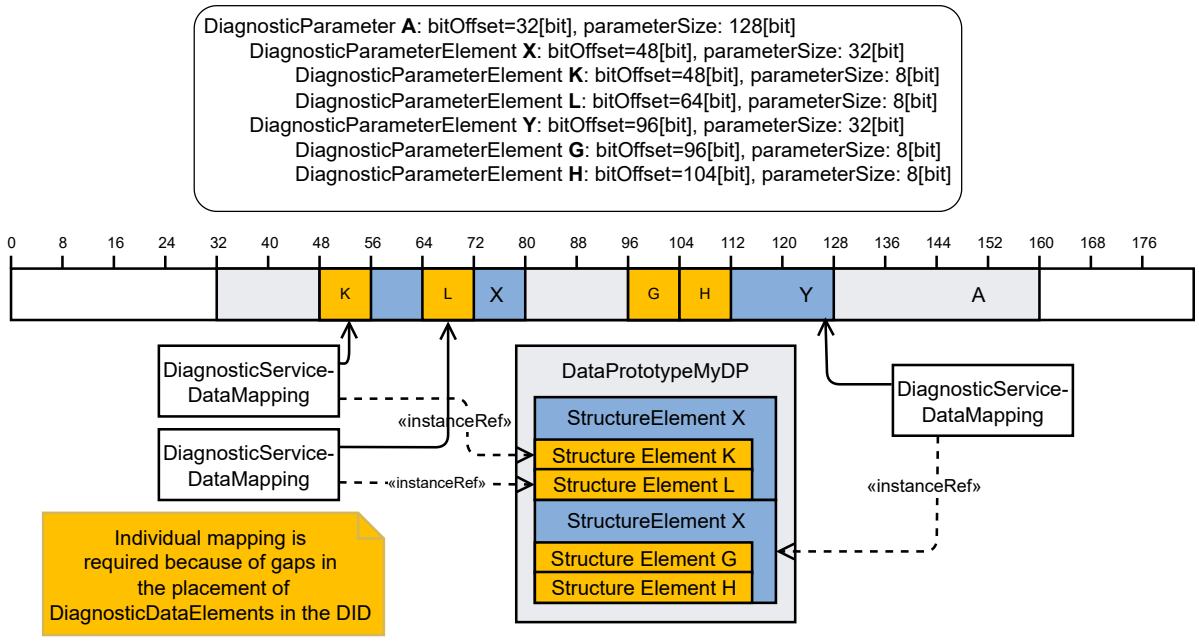

Figure B.7: Example for the mapping of a nested structure to a `DiagnosticParameterElement`

B.2.3 Array of Structure

The next example (as depicted in Figure B.8) explains how the mapping to array of composite data objects works.

More specifically, the mapping example corresponds to the example discussed in Chapter B.1.4, resp. Figure B.4.

In this example, the content of the first array element in the example (X[0]) has been mapped element by element (i.e. “R”, “G”, and “B”) for diagnostics purposes only.

It would also be possible to map the entire array elements, as indicated by the mapping of X[1] and X[2].

The individual mapping of an element of the structure is explained in more detail within Figure B.8[1].

```
<DIAGNOSTIC-SERVICE-DATA-MAPPING>
  <SHORT-NAME>R0ToR0</SHORT-NAME>
  <MAPPED-DATA-ELEMENT-IREF>
    <CONTEXT-PORT-REF DEST="P-PORT-PROTOTYPE">/SwComponentTypes/MySwc/RgbArrayPort</CONTEXT-
        PORT-REF>
    <CONTEXT-DATA-PROTOTYPE-REF DEST="APPLICATION-ARRAY-ELEMENT" INDEX="0">/
        ApplicationDataTypes/ArrayOfRgbStructType/arrayElement</CONTEXT-DATA-PROTOTYPE-REF>
    <TARGET-DATA-PROTOTYPE-REF DEST="APPLICATION-RECORD-ELEMENT">/ApplicationDataTypes/
        RgbStructType/R</TARGET-DATA-PROTOTYPE-REF>
  </MAPPED-DATA-ELEMENT-IREF>
  <PARAMETER-ELEMENT-ACCESS>
```

[1]Please note that the InstanceRef for the identification of the `DataPrototype` in the application software intentionally starts at the level of `PortPrototype` to keep things simple and the complexity of the example model only as high as needed for this purpose.

```
    <CONTEXT-ELEMENT-REFS>
      <CONTEXT-ELEMENT-REF DEST="DIAGNOSTIC-PARAMETER-ELEMENT" INDEX="0">/
          DiagnosticDataElements/DidArrayOfStruct/A/X</CONTEXT-ELEMENT-REF>
    </CONTEXT-ELEMENT-REFS>
    <TARGET-ELEMENT-REF DEST="DIAGNOSTIC-PARAMETER-ELEMENT">/DiagnosticDataElements/
       DidArrayOfStruct/A/X/R</TARGET-ELEMENT-REF>
  </PARAMETER-ELEMENT-ACCESS>
</DIAGNOSTIC-SERVICE-DATA-MAPPING>
```

Listing B.5: Example for the mapping of an element of a structure that itself represents an array element

Please note that in this specific case **two** references to `DiagnosticParameter-Element` are needed to properly identify the mapped `DiagnosticParameter-Element`.

The first reference (in the role `DiagnosticServiceDataMapping.parameter-ElementAccess.contextElement`) is used to identify the array element (note the appearance of the `index` XML attribute).

The second reference (in the role `DiagnosticServiceDataMapping.parameter-ElementAccess.targetElement`) identifies the element of the structure.

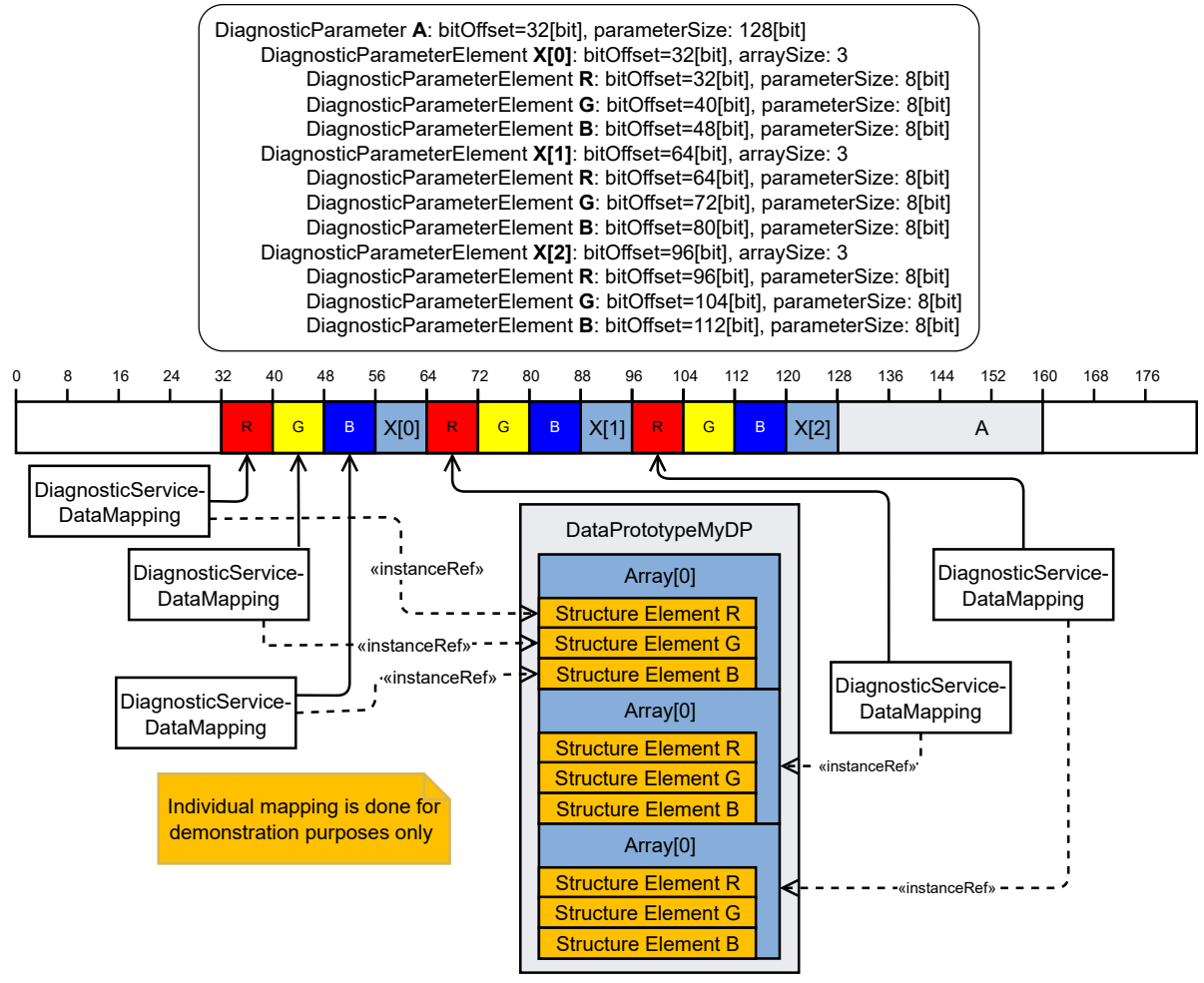

Figure B.8: Example for the mapping of a `DiagnosticParameter` that represents an array of structures

The mapping of the entire array elements is sketched in Listing B.6.

Obviously, in this case only a singe reference (in the role DiagnosticService-DataMapping.parameterElementAccess.targetElement) to a Diagnostic-ParameterElement – including the index XML attribute, in this case set to "1" – is needed.

```
<DIAGNOSTIC-SERVICE-DATA-MAPPING>
  <SHORT-NAME>X1toArray1</SHORT-NAME>
  <MAPPED-DATA-ELEMENT-IREF>
    <CONTEXT-PORT-REF DEST="P-PORT-PROTOTYPE">/SwComponentTypes/MySwc/RgbArrayPort</CONTEXT-
        PORT-REF>
    <TARGET-DATA-PROTOTYPE-REF DEST="APPLICATION-ARRAY-ELEMENT" INDEX="1">/
        ApplicationDataTypes/ArrayOfRgbStructType/arrayElement</TARGET-DATA-PROTOTYPE-REF>
  </MAPPED-DATA-ELEMENT-IREF>
  <PARAMETER-ELEMENT-ACCESS>
    <TARGET-ELEMENT-REF DEST="DIAGNOSTIC-PARAMETER-ELEMENT" INDEX="1">/DiagnosticDataElements/
        DidArrayOfStruct/A/X</TARGET-ELEMENT-REF>
  </PARAMETER-ELEMENT-ACCESS>
</DIAGNOSTIC-SERVICE-DATA-MAPPING>
```

Listing B.6: Example for the mapping of an entire array element

B.3 Examples of the Definition of Mature and De-Mature Condition

The following ARXML fragment exemplifies the usage of StructuredReq along with the standardized values of the attribute category to attach semi-formal textual descriptions of an exemplary mature condition and a de-mature condition to a DiagnosticEvent according to [TPS_DEXT_01069].

```
<DIAGNOSTIC-EVENT>
  <SHORT-NAME>ExampleEvent_0001</SHORT-NAME>
  <INTRODUCTION>
    <STRUCTURED-REQ>
      <SHORT-NAME>MatureCondition</SHORT-NAME>
      <CATEGORY>DIAG_EVENT_MAT_COND</CATEGORY>
      <DESCRIPTION>
        <P>
          <L-1 L="EN">This DTC is set if System Voltage is below 9 Volts</L-1>
        </P>
      </DESCRIPTION>
    </STRUCTURED-REQ>
    <STRUCTURED-REQ>
      <SHORT-NAME>DematureCondition</SHORT-NAME>
      <CATEGORY>DIAG_EVENT_DEMAT_COND</CATEGORY>
      <DESCRIPTION>
        <P>
          <L-1 L="EN">This DTC is set if System Voltage is above 10 Volts<XFILE>
            <SHORT-NAME>Requirement_Specification</SHORT-NAME>
            <URL>http://autosar.org</URL>
          </XFILE>
        </L-1>
      </P>
    </DESCRIPTION>
  </STRUCTURED-REQ>
  </INTRODUCTION>
  <EVENT-CLEAR-ALLOWED>ALWAYS</EVENT-CLEAR-ALLOWED>
  <EVENT-KIND>SWC</EVENT-KIND>
  <PRESTORAGE-FREEZE-FRAME>false</PRESTORAGE-FREEZE-FRAME>
</DIAGNOSTIC-EVENT>
```

Listing B.7: Example for the definition of a semi-formal textual element in the context of a DiagnosticEvent

B.4 Examples of the Definition of semi-formal Textual Description

The following ARXML fragment exemplifies the usage of TraceableText along with the standardized values of the attribute category to attach semi-formal textual descriptions to a DiagnosticTroubleCodeUds. For more information, please refer to [TPS_DEXT_01066].

```
<DIAGNOSTIC-TROUBLE-CODE-UDS>
  <SHORT-NAME>ExampleDTC_0001</SHORT-NAME>
  <LONG-NAME>
    <L-4 L="EN">My little ODX long name</L-4>
  </LONG-NAME>
  <DESC>
    <L-2 L="EN">This DTC is a System Error DTC</L-2>
  </DESC>
  <INTRODUCTION>
    <TRACE>
      <SHORT-NAME>MyErrorText</SHORT-NAME>
      <CATEGORY>DIAG_DTC_ERROR_TEXT</CATEGORY>
      <P>
        <L-1 L="LA">Lorem ipsum dolor sit amet, consectetur adipisicing elit</L-1>
      </P>
    </TRACE>
  </INTRODUCTION>
  <FUNCTIONAL-UNIT>1</FUNCTIONAL-UNIT>
  <SEVERITY>CHECK-AT-NEXT-HALT</SEVERITY>
  <TROUBLE-CODE-PROPSS>
    <DIAGNOSTIC-TROUBLE-CODE-PROPS-REF-CONDITIONAL>
      <DIAGNOSTIC-TROUBLE-CODE-PROPS-REF DEST="DIAGNOSTIC-TROUBLE-CODE-PROPS">/AUTOSAR/
        UseCase_230/ExampleDTC_0001_Props</DIAGNOSTIC-TROUBLE-CODE-PROPS-REF>
    </DIAGNOSTIC-TROUBLE-CODE-PROPS-REF-CONDITIONAL>
  </TROUBLE-CODE-PROPSS>
  <UDS-DTC-VALUE>0x000001</UDS-DTC-VALUE>
</DIAGNOSTIC-TROUBLE-CODE-UDS>
```

Listing B.8: Example for the definition of a semi-formal textual element in the context of a DiagnosticTroubleCode

B.5 Examples of the Definition of Pre- and Run-Condition

The following ARXML fragment exemplifies the usage of StructuredReq along with the standardized values of the attribute category to attach semi-formal textual descriptions to a DiagnosticAccessPermission. For more information, please refer to [TPS_DEXT_01071].

```
<DIAGNOSTIC-ACCESS-PERMISSION>
  <SHORT-NAME>exampleAccessPermission</SHORT-NAME>
  <INTRODUCTION>
    <STRUCTURED-REQ>
      <SHORT-NAME>precondition</SHORT-NAME>
      <CATEGORY>DIAG_ACCESS_PERM_PRE_COND</CATEGORY>
      <DESCRIPTION>
        <P>
          <L-1 L="EN">This is a textual description of a pre-condition</L-1>
        </P>
      </DESCRIPTION>
    </STRUCTURED-REQ>
    <STRUCTURED-REQ>
      <SHORT-NAME>runcondition</SHORT-NAME>
      <CATEGORY>DIAG_ACCESS_PERM_RUN_COND</CATEGORY>
      <DESCRIPTION>
        <P>
          <L-1 L="EN">This is a textual description of a run-condition</L-1>
```

```
        </P>
      </DESCRIPTION>
    </STRUCTURED-REQ>
  </INTRODUCTION>
  <DIAGNOSTIC-SESSION-REFS>
    <DIAGNOSTIC-SESSION-REF DEST="DIAGNOSTIC-SESSION">/AUTOSAR/UseCase_230/ExampleSession</
        DIAGNOSTIC-SESSION-REF>
  </DIAGNOSTIC-SESSION-REFS>
  <SECURITY-LEVEL-REFS>
    <SECURITY-LEVEL-REF DEST="DIAGNOSTIC-SECURITY-LEVEL">/AUTOSAR/UseCase_230/
        ExampleSecurityLevel</SECURITY-LEVEL-REF>
  </SECURITY-LEVEL-REFS>
</DIAGNOSTIC-ACCESS-PERMISSION>
```

Listing B.9: Example for the definition of pre- and run-conditions for `DiagnosticAccessPermission`

C Mentioned Class Tables

For the sake of completeness, this chapter contains a set of class tables representing meta-classes mentioned in the context of this document but which are not contained directly in the scope of describing specific meta-model semantics.

Class	***ARElement*** (abstract)			
Note	An element that can be defined stand-alone, i.e. without being part of another element (except for packages of course).			
Base	*ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	*AbstractMessageToSecureChannelMapping*, AclObjectSet, AclOperation, AclPermission, AclRole, AliasNameSet, ApplicabilityInfoSet, ApplicationPartition, *AutosarDataType*, *BaseType*, BlueprintMappingSet, BswEntryRelationshipSet, BswModuleDescription, BswModuleEntry, BuildActionManifest, CalibrationParameterValueSet, ClientIdDefinitionSet, Collection, CompuMethod, ConsistencyNeedsBlueprintSet, ConstantSpecification, ConstantSpecificationMappingSet, CpSoftwareCluster, CpSoftwareClusterBinaryManifestDescriptor, CpSoftwareClusterMappingSet, CpSoftwareClusterResourcePool, CryptoEllipticCurveProps, CryptoServiceCertificate, CryptoServiceKey, CryptoServicePrimitive, CryptoServiceQueue, CryptoSignatureScheme, DataConstr, DataTransformationSet, DataTypeMappingSet, DdsCpConfig, *DiagnosticCommonElement*, DiagnosticConnection, DiagnosticContributionSet, DltArgumentPropsSet, DltContext, DltEcu, Documentation, E2EProfileCompatibilityProps, EcucDefinitionCollection, EcucDestinationUriDefSet, EcucModuleConfigurationValues, EcucModuleDef, EcucValueCollection, EthIpProps, EthTcpIpIcmpProps, EthTcpIpProps, EvaluatedVariantSet, FMFeature, FMFeatureMap, FMFeatureModel, FMFeatureSelectionSet, FirewallRule, FlatMap, GeneralPurposeConnection, HwCategory, HwElement, HwType, *IEEE1722TpConnection*, IPSecConfigProps, IPv6ExtHeaderFilterSet, *IdsCommonElement*, IdsDesign, *Implementation*, ImpositionTimeDefinitionGroup, InterpolationRoutineMappingSet, J1939ControllerApplication, KeywordSet, LifeCycleInfoSet, LifeCycleStateDefinitionGroup, LogAndTraceMessageCollectionSet, MacSecGlobalKayProps, MacSecParticipantSet, McFunction, McGroup, ModeDeclarationGroup, ModeDeclarationMappingSet, OsTaskProxy, PhysicalDimension, PhysicalDimensionMappingSet, *PortInterface*, PortInterfaceMappingSet, PortPrototypeBlueprint, PostBuildVariantCriterion, PostBuildVariantCriterionValueSet, PredefinedVariant, RapidPrototypingScenario, SdgDef, *SecureComProps*, *SecurityConnectivityAssociation*, SignalServiceTranslationPropsSet, SomeipSdClientEventGroupTimingConfig, SomeipSdClientServiceInstanceConfig, SomeipSdServerEventGroupTimingConfig, SomeipSdServerServiceInstanceConfig, SwAddrMethod, SwAxisType, SwComponentMappingConstraints, *SwComponentType*, SwRecordLayout, SwSystemconst, SwSystemconstantValueSet, SwcBswMapping, System, SystemComSpecDefinitionSet, SystemSignal, SystemSignalGroup, TDCpSoftwareClusterMappingSet, TcpOptionFilterSet, *TimingExtension*, TlsConnectionGroup, TlvDataIdDefinitionSet, TransformationPropsSet, Unit, UnitGroup, *UploadablePackageElement*, ViewMapSet			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table C.1: ARElement

Class	**ARPackage**			
Note	AUTOSAR package, allowing to create top level packages to structure the contained ARElements. ARPackages are open sets. This means that in a file based description system multiple files can be used to partially describe the contents of a package. This is an extended version of MSR's SW-SYSTEM.			
Base	*ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ARPackage.arPackage, AUTOSAR.arPackage			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**ARPackage**			
arPackage	ARPackage	*	aggr	This represents a sub package within an ARPackage, thus allowing for an unlimited package hierarchy. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=arPackage, arPackage.shortName, ar Package.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30
element	PackageableElement	*	aggr	Elements that are part of this package **Stereotypes:** atpVariation **Tags:** atp.Splitkey=element, element.shortName, element.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=20
referenceBase	ReferenceBase	*	aggr	This denotes the reference bases for the package. This is the basis for all relative references within the package. The base needs to be selected according to the base attribute within the references. **Tags:** atp.Splitkey=referenceBase, referenceBase.shortLabel xml.sequenceOffset=10

Table C.2: ARPackage

Class	**AUTOSAR**			
Note	Root element of an AUTOSAR description, also the root element in corresponding XML documents. **Tags:** xml.globalElement=true			
Base	*ARObject*			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
adminData	AdminData	0..1	aggr	This represents the administrative data of an Autosar file. **Tags:** atp.Splitkey=adminData xml.sequenceOffset=10
arPackage	ARPackage	*	aggr	This is the top level package in an AUTOSAR model. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=arPackage, arPackage.shortName, ar Package.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30
fileInfo Comment	FileInfoComment	0..1	aggr	This represents a possibility to provide a structured comment in an AUTOSAR file. **Stereotypes:** atpStructuredComment **Tags:** atp.Splitkey=fileInfoComment xml.roleElement=true xml.sequenceOffset=-10 xml.typeElement=false

▽

△

Class	**AUTOSAR**			
introduction	DocumentationBlock	0..1	aggr	This represents an introduction on the Autosar file. It is intended for example to represent disclaimers and legal notes. **Tags:** atp.Splitkey=introduction xml.sequenceOffset=20

Table C.3: AUTOSAR

Class	***ApplicationDataType*** (abstract)			
Note	`ApplicationDataType` defines a data type from the application point of view. Especially it should be used whenever something "physical" is at stake. An `ApplicationDataType` represents a set of values as seen in the application model, such as measurement units. It does not consider implementation details such as bit-size, endianess, etc. It should be possible to model the application level aspects of a VFB system by using `ApplicationDataType`s only.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	*ApplicationCompositeDataType*, ApplicationPrimitiveDataType			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table C.4: ApplicationDataType

Class	**ApplicationValueSpecification**			
Note	This meta-class represents values for DataPrototypes typed by ApplicationDataTypes (this includes in particular compound primitives). For further details refer to ASAM CDF 2.0. This meta-class corresponds to some extent with SW-INSTANCE in ASAM CDF 2.0.			
Base	*ARObject*, *CompositeRuleBasedValueArgument*, *ValueSpecification*			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, CompositeRuleBasedValueSpecification.compoundPrimitive Argument, ConstantSpecification.valueSpec, CryptoServiceKey.developmentValue, DiagnosticEnvData Condition.compareValue, DiagnosticEnvDataElementCondition.compareValue, DiagnosticEnvSovdData Condition.compareValue, FieldSenderComSpec.initValue, ISignal.initValue, ISignal.timeoutSubstitution Value, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.rom BlockInitValue, NvRequireComSpec.initValue, ParameterDataPrototype.initValue, ParameterProvideCom Spec.initValue, ParameterRequireComSpec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, PortDefinedArgumentValue.value, PortPrototypeBlueprintInit Value.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, Sw DataDefProps.invalidValue, VariableDataPrototype.initValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
category	Identifier	0..1	attr	Specifies to which category of ApplicationDataType this ApplicationValueSpecification can be applied (e.g. as an initial value), thus imposing constraints on the structure and semantics of the contained values, see [constr_1006] and [constr_1519].

▽

△

Class	**ApplicationValueSpecification**			
swAxisCont (ordered)	SwAxisCont	*	aggr	This represents the axis values of a Compound Primitive Data Type (curve or map). The first swAxisCont describes the x-axis, the second sw AxisCont describes the y-axis, the third swAxisCont describes the z-axis. In addition to this, the axis can be denoted in swAxisIndex.
swValueCont	SwValueCont	0..1	aggr	This represents the values of a Compound Primitive Data Type. **Tags:** atp.Splitkey=swValueCont

Table C.5: ApplicationValueSpecification

Class	**ArgumentDataPrototype**			
Note	An argument of an operation, carries direction and implementation information.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *AutosarDataPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, ClientServerOperation.argument			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
direction	ArgumentDirectionEnum	0..1	attr	This attribute specifies the direction of the argument.
serverArgumentImplPolicy	ServerArgumentImplPolicyEnum	0..1	attr	This defines how the argument type of the servers `RunnableEntity` is implemented. If the attribute is not defined this has the same semantics as if the attribute is set to the value `useArgumentType` for primitive arguments and structures.

Table C.6: ArgumentDataPrototype

Class	***AtomicSwComponentType*** (abstract)			
Note	An atomic software component is atomic in the sense that it cannot be further decomposed and distributed across multiple ECUs.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Subclasses	ApplicationSwComponentType, ComplexDeviceDriverSwComponentType, EcuAbstractionSwComponentType, NvBlockSwComponentType, SensorActuatorSwComponentType, ServiceProxySwComponentType, ServiceSwComponentType			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
internalBehavior	SwcInternalBehavior	0..1	aggr	The `SwcInternalBehavior`s owned by an `AtomicSwComponentType` can be located in a different physical file. Therefore the aggregation is <<atp Splitable>>. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=internalBehavior, internalBehavior.shortName, internalBehavior.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the `SymbolProps` for the `AtomicSwComponentType`. **Tags:** atp.Splitkey=symbolProps, symbolProps.shortName

Table C.7: AtomicSwComponentType

Class	***AtpInstanceRef*** (abstract)			
Note	An M0 instance of a classifier may be represented as a tree rooted at that instance, where under each node come the sub-trees representing the instances which act as features under that node. An instance ref specifies a navigation path from any M0 tree-instance of the base (which is a classifier) to a leaf (which is an instance of the target).			
Base	*ARObject*			
Subclasses	AnyInstanceRef, ApplicationCompositeElementInPortInterfaceInstanceRef, ComponentInComposition InstanceRef, ComponentInSystemInstanceRef, *DataPrototypeInPortInterfaceInstanceRef*, Data PrototypeInSystemInstanceRef, InnerDataPrototypeGroupInCompositionInstanceRef, InnerPortGroupIn CompositionInstanceRef, InnerRunnableEntityGroupInCompositionInstanceRef, InstanceEventIn CompositionInstanceRef, ModeDeclarationGroupPrototypeInSystemInstanceRef, *ModeGroupInAtomic SwcInstanceRef*, ModeInBswInstanceRef, ModeInBswModuleDescriptionInstanceRef, ModeInSwc InstanceRef, OperationArgumentInComponentInstanceRef, *OperationInAtomicSwcInstanceRef*, OperationInSystemInstanceRef, PModeInSystemInstanceRef, ParameterDataPrototypeInSystem InstanceRef, ParameterInAtomicSWCTypeInstanceRef, PortGroupInSystemInstanceRef, *PortIn CompositionTypeInstanceRef*, RModeInAtomicSwcInstanceRef, RteEventInCompositionInstanceRef, RteEventInEcuInstanceRef, RteEventInSystemInstanceRef, RunnableEntityInCompositionInstanceRef, SwcServiceDependencyInSystemInstanceRef, *TriggerInAtomicSwcInstanceRef*, TriggerInSystem InstanceRef, VariableAccessInEcuInstanceRef, VariableDataPrototypeInCompositionInstanceRef, VariableDataPrototypeInSystemInstanceRef, VariableInAtomicSWCTypeInstanceRef, *VariableInAtomic SwcInstanceRef*, VariableInComponentInstanceRef			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
atpBase	AtpClassifier	0..1	ref	This is the base from which the navigaion path starts. **Stereotypes:** atpAbstract; atpDerived
atpContext Element (ordered)	AtpPrototype	*	ref	This is one particular step in the navigation path. **Stereotypes:** atpAbstract
atpTarget	AtpFeature	0..1	ref	This is the target of the instance ref. In other words it is the terminal of the navigation path. **Stereotypes:** atpAbstract

Table C.8: AtpInstanceRef

Class	***AutosarDataPrototype*** (abstract)			
Note	Base class for prototypical roles of an `AutosarDataType`.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	ArgumentDataPrototype, ParameterDataPrototype, VariableDataPrototype			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
type	AutosarDataType	0..1	tref	This represents the corresponding data type. **Stereotypes:** isOfType **Tags:** atp.Splitkey=type

Table C.9: AutosarDataPrototype

Class	***AutosarDataType*** (abstract)			
Note	Abstract base class for user defined AUTOSAR data types for software.			
Base	*ARElement*, *ARObject*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Subclasses	*AbstractImplementationDataType*, *ApplicationDataType*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
swDataDef Props	SwDataDefProps	0..1	aggr	The properties of this AutosarDataType. **Tags:** atp.Splitkey=swDataDefProps

Table C.10: AutosarDataType

Class	**BaseType** (abstract)			
Note	This abstract meta-class represents the ability to specify a platform dependent base type.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	SwBaseType			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
baseType Definition	BaseTypeDefinition	1	aggr	This is the actual definition of the base type. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table C.11: BaseType

Class	**BaseTypeDirectDefinition**			
Note	This BaseType is defined directly (as opposite to a derived BaseType)			
Base	*ARObject*, *BaseTypeDefinition*			
Aggregated by	*BaseType*.baseTypeDefinition			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
baseType Encoding	BaseTypeEncoding String	0..1	attr	This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **Tags:** xml.sequenceOffset=90
baseTypeSize	PositiveInteger	0..1	attr	Describes the length of the data type specified in the container in bits. **Tags:** xml.sequenceOffset=70
byteOrder	ByteOrderEnum	0..1	attr	This attribute specifies the byte order of the base type. **Tags:** xml.sequenceOffset=110
memAlignment	PositiveInteger	0..1	attr	This attribute describes the alignment of the memory object in bits. E.g. "8" specifies, that the object in question is aligned to a byte while "32" specifies that it is aligned four byte. If the value is set to "0" the meaning shall be interpreted as "unspecified". **Tags:** xml.sequenceOffset=100
native Declaration	NativeDeclarationString	0..1	attr	This attribute describes the declaration of such a base type in the native programming language, primarily in the Programming language C. This can then be used by a code generator to include the necessary declarations into a header file. For example BaseType with shortName: "MyUnsignedInt" native Declaration: "unsigned short" Results in typedef unsigned short MyUnsignedInt; If the attribute is not defined the referring Implementation DataTypes will not be generated as a typedef by RTE. If a nativeDeclaration type is given it shall fulfill the characteristic given by basetypeEncoding and baseType Size. ▽

▽

△

Class	BaseTypeDirectDefinition			
				△ This is required to ensure the consistent handling and interpretation by software components, RTE, COM and MCM systems. **Tags:** xml.sequenceOffset=120

Table C.12: BaseTypeDirectDefinition

Class	BswModuleDescription			
Note	Root element for the description of a single BSW module or BSW cluster. In case it describes a BSW module, the short name of this element equals the name of the BSW module. **Tags:** atp.recommendedPackage=BswModuleDescriptions This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element, *AtpClassifier*.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
bswModule Dependency	BswModuleDependency	*	aggr	Describes the dependency to another BSW module. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=bswModuleDependency, bswModule Dependency.shortName, bswModule Dependency.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=20
bswModule Documentation	SwComponent Documentation	0..1	aggr	This adds a documentation to the BSW module. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=bswModuleDocumentation, bswModule Documentation.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=6
expectedEntry	BswModuleEntry	*	ref	Indicates an entry which is required by this module. Replacement of outgoingCallback / requiredEntry. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=expectedEntry, expectedEntry.bswModule Entry, expectedEntry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
implemented Entry	BswModuleEntry	*	ref	Specifies an entry provided by this module which can be called by other modules. This includes "main" functions, interrupt routines, and callbacks. Replacement of providedEntry / expectedCallback. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=implementedEntry, implementedEntry.bsw ModuleEntry, implementedEntry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**BswModuleDescription**			
internalBehavior	BswInternalBehavior	*	aggr	The various BswInternalBehaviors associated with a Bsw ModuleDescription can be distributed over several physical files. Therefore the aggregation is <<atp Splitable>>. **Tags:** atp.Splitkey=internalBehavior, internalBehavior.short Name xml.sequenceOffset=65
moduleId	PositiveInteger	0..1	attr	Refers to the BSW Module Identifier defined by the AUTOSAR standard. For non-standardized modules, a proprietary identifier can be optionally chosen. **Tags:** xml.sequenceOffset=5
providedClient ServerEntry	BswModuleClientServer Entry	*	aggr	Specifies that this module provides a client server entry which can be called from another partition or core.This entry is declared locally to this context and will be connected to the requiredClientServerEntry of another or the same module via the configuration of the BSW Scheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=providedClientServerEntry, providedClient ServerEntry.shortName, providedClientServer Entry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=45
providedData	VariableDataPrototype	*	aggr	Specifies a data prototype provided by this module in order to be read from another partition or core.The providedData is declared locally to this context and will be connected to the requiredData of another or the same module via the configuration of the BSW Scheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=providedData, providedData.shortName, providedData.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=55
providedMode Group	ModeDeclarationGroup Prototype	*	aggr	A set of modes which is owned and provided by this module or cluster. It can be connected to the required ModeGroups of other modules or clusters via the configuration of the BswScheduler. It can also be synchronized with modes provided via ports by an associated ServiceSwComponentType, EcuAbstraction SwComponentType or ComplexDeviceDriverSw ComponentType. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=providedModeGroup, providedMode Group.shortName, providedModeGroup.variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=25

▽

△

Class	**BswModuleDescription**			
releasedTrigger	Trigger	*	aggr	A Trigger released by this module or cluster. It can be connected to the requiredTriggers of other modules or clusters via the configuration of the BswScheduler. It can also be synchronized with Triggers provided via ports by an associated ServiceSwComponentType, Ecu AbstractionSwComponentType or ComplexDeviceDriver SwComponentType. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=releasedTrigger, releasedTrigger.shortName, releasedTrigger.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=35
requiredClient ServerEntry	BswModuleClientServer Entry	*	aggr	Specifies that this module requires a client server entry which can be implemented on another partition or core.This entry is declared locally to this context and will be connected to the providedClientServerEntry of another or the same module via the configuration of the BSW Scheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=requiredClientServerEntry, requiredClient ServerEntry.shortName, requiredClientServer Entry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=50
requiredData	VariableDataPrototype	*	aggr	Specifies a data prototype required by this module in oder to be provided from another partition or core.The required Data is declared locally to this context and will be connected to the providedData of another or the same module via the configuration of the BswScheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=requiredData, requiredData.shortName, requiredData.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=60
requiredMode Group	ModeDeclarationGroup Prototype	*	aggr	Specifies that this module or cluster depends on a certain mode group. The requiredModeGroup is local to this context and will be connected to the providedModeGroup of another module or cluster via the configuration of the BswScheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=requiredModeGroup, requiredMode Group.shortName, requiredModeGroup.variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=30

▽

△

Class	BswModuleDescription			
requiredTrigger	Trigger	*	aggr	Specifies that this module or cluster reacts upon an external trigger.This requiredTrigger is declared locally to this context and will be connected to the providedTrigger of another module or cluster via the configuration of the BswScheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=requiredTrigger, requiredTrigger.shortName, requiredTrigger.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=40

Table C.13: BswModuleDescription

Class	ClientServerOperation			
Note	An operation declared within the scope of a client/server interface.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ApplicationInterface.command, *AtpClassifier*.atpFeature, ClientServerInterface.operation, DiagnosticDataElementInterface.read, DiagnosticDataIdentifierInterface.read, DiagnosticDataIdentifierInterface.write, DiagnosticExtendedDataRecordInterface.provide, DiagnosticRoutineInterface.requestResult, DiagnosticRoutineInterface.start, DiagnosticRoutineInterface.stop, ServiceInterface.method			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
argument (ordered)	ArgumentDataPrototype	*	aggr	An argument of this `ClientServerOperation`. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=blueprintDerivationTime
diagArgIntegrity	Boolean	0..1	attr	This attribute shall only be used in the implementation of diagnostic routines to support the case where input and output arguments are allocated in a shared buffer and might unintentionally overwrite input arguments by tentative write operations to output arguments. This situation can happen during sliced execution or while output parameters are arrays (call by reference). The value true means that the `ClientServerOperation` is aware of the usage of a shared buffer and takes precautions to avoid unintentional overwrite of input arguments. If the attribute does not exist or is set to false the `ClientServerOperation` does not have to consider the usage of a shared buffer. This Attribute is only used by the AUTOSAR Classic Platform.
possibleError	ApplicationError	*	ref	Possible errors that may by raised by the referring operation. **Tags:** atp.Splitkey=possibleError This Attribute is only used by the AUTOSAR Classic Platform.

Table C.14: ClientServerOperation

Class	***CommConnectorPort*** (abstract)			
Note	The Ecu communication relationship defines which signals, Pdus and frames are actually received and transmitted by this ECU. For each signal, Pdu or Frame that is transmitted or received and used by the Ecu an association between an ISignalPort, IPduPort or FramePort with the corresponding Triggering shall be created. An ISignalPort shall be created only if the corresponding signal is handled by COM (RTE or Signal Gateway). If a Pdu Gateway ECU only routes the Pdu without being interested in the content only a FramePort and an IPduPort needs to be created.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	FramePort, IPduPort, ISignalPort			
Aggregated by	*CommunicationConnector*.ecuCommPortInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Direction	Communication DirectionType	0..1	attr	Communication Direction of the Connector Port (input or output Port).

Table C.15: CommConnectorPort

Enumeration	**CommunicationDirectionType**
Note	Describes the communication direction.
Aggregated by	*CommConnectorPort*.communicationDirection, *IEEE1722TpConnection*.communicationDirection, IPSecRule.direction, ISignalIPduGroup.communicationDirection
Literal	***Description***
in	Reception (Input) **Tags:** atp.EnumerationLiteralIndex=0
out	Transmission (Output) **Tags:** atp.EnumerationLiteralIndex=1

Table C.16: CommunicationDirectionType

Class	**CompositionSwComponentType**			
Note	A `CompositionSwComponentType` aggregates `SwComponentPrototype`s (that in turn are typed by `SwComponentType`)s as well as `SwConnector`s for primarily connecting `SwComponentPrototype`s among each others and towards the surface of the `CompositionSwComponentType`. By this means, a hierarchical structures of software-components can be created. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**CompositionSwComponentType**			
component	SwComponent Prototype	*	aggr	The instantiated components that are part of this composition. The aggregation of SwComponentPrototype is subject to variability with the purpose to support the conditional existence of a SwComponentPrototype. Please be aware: if the conditional existence of SwComponentPrototypes is resolved post-build, the deselected SwComponentPrototypes are still contained in the ECUs build but the instances are inactive in that they are not scheduled by the RTE. The use case for having 0 components owned by the CompositionSwComponentType could be to deliver an empty CompositionSwComponentType to e.g. a supplier for filling the internal structure. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=component, component.shortName, component.variationPoint.shortLabel vh.latestBindingTime=postBuild
connector	SwConnector	*	aggr	SwConnectors have the principal ability to establish a connection among PortPrototypes. They can have many roles in the context of a CompositionSwComponentType. Details are refined by subclasses. The aggregation of SwConnectors is subject to variability with the purpose to support variant data flow. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=connector, connector.shortName, connector.variationPoint.shortLabel vh.latestBindingTime=postBuild
constantValue Mapping	ConstantSpecification MappingSet	*	ref	Reference to the ConstantSpecificationMapping to be applied for initValues of PPortComSpecs and RPortComSpec. **Tags:** atp.Splitkey=constantValueMapping
dataType Mapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMappingSet to be applied for the used ApplicationDataTypes in PortInterfaces. Background: when developing subsystems it may happen that ApplicationDataTypes are used on the surface of CompositionSwComponentTypes. In this case it would be reasonable to be able to also provide the intended mapping to the ImplementationDataTypes. However, this mapping shall be informal and not technically binding for the implementors mainly because the RTE generator is not concerned about the CompositionSwComponentTypes. Rationale: if the mapping of ApplicationDataTypes on the delegated and inner PortPrototype matches then the mapping to ImplementationDataTypes is not impacting compatibility. **Tags:** atp.Splitkey=dataTypeMapping

▽

△

Class	**CompositionSwComponentType**			
instantiation RTEEventProps	InstantiationRTEEvent Props	*	aggr	This allows to define instantiation specific properties for RTE Events, in particular for instance specific scheduling. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=instantiationRTEEventProps, instantiation RTEEventProps.shortLabel, instantiationRTEEvent Props.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime This Attribute is only used by the AUTOSAR Classic Platform.
physical Dimension Mapping	PhysicalDimension MappingSet	0..1	ref	This reference identifies the `PhysicalDimensionMappingSet` that is applicable in the context of the enclosing `CompositionSwComponentType`. The `PhysicalDimensionMapping`s contained in the `PhysicalDimensionMappingSet` shall be taken into account for the assessment of the compatibility of `PhysicalDimension`s in the context of creation of a `PortInterfaceMapping` in the scope of the `CompositionSwComponentType`. **Tags:** atp.Splitkey=physicalDimensionMapping

Table C.17: CompositionSwComponentType

Class	**CompuMethod**			
Note	This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant. **Tags:** atp.recommendedPackage=CompuMethods			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
compuInternal ToPhys	Compu	0..1	aggr	This specifies the computation from internal values to physical values. **Tags:** atp.Splitkey=compuInternalToPhys xml.sequenceOffset=80
compuPhysTo Internal	Compu	0..1	aggr	This represents the computation from physical values to the internal values. **Tags:** atp.Splitkey=compuPhysToInternal xml.sequenceOffset=90
displayFormat	DisplayFormatString	0..1	attr	This property specifies, how the physical value shall be displayed e.g. in documents or measurement and calibration tools. **Tags:** xml.sequenceOffset=20
unit	Unit	0..1	ref	This is the physical unit of the Physical values for which the CompuMethod applies. **Tags:** atp.Splitkey=unit xml.sequenceOffset=30

Table C.18: CompuMethod

Class	CpSoftwareCluster			
Note	This meta class provides the ability to define a CP Software Cluster. Each CP Software Cluster can be integrated and build individually. It defines the sub-set of hierarchical tree(s) of Software Components belonging to this CP Software Cluster. Resources required or provided by this CP Software Cluster are given in the according mappings. **Tags:** atp.recommendedPackage=CpSoftwareClusters This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
softwareCluster Id	PositiveInteger	0..1	attr	This attribute represents the value of the id of the corresponding CP software cluster.
swComponent Assignment	SwComponent PrototypeAssignment	*	aggr	This is the collection of SwComponentPrototype Assignments **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swComponentAssignment, swComponent Assignment.variationPoint.shortLabel vh.latestBindingTime=postBuild
swComposition	CompositionSw ComponentType	*	ref	Software Components in the context of a CompositionSw ComponentType belonging to this CP Software Cluster. This reference can be used to describe the belonging SWCs when the CP Software Cluster is described out of the context of a System, e.g. reusable CP Software Cluster. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swComposition, swComposition.composition SwComponentType, swComposition.variationPoint.short Label vh.latestBindingTime=systemDesignTime

Table C.19: CpSoftwareCluster

Class	CryptoServiceCertificate			
Note	This meta-class represents the ability to model a cryptographic certificate. **Tags:** atp.recommendedPackage=CryptoServiceCertificates			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
algorithmFamily	CryptoCertificate AlgorithmFamilyEnum	0..1	attr	This attribute represents a description of the family of crypto algorithm used to generate public key and signature of the cryptographic certificate.
format	CryptoCertificateFormat Enum	0..1	attr	This attribute can be used to provide information about the format used to create the certificate
maximum Length	PositiveInteger	0..1	attr	This attribute represents the ability to define the maximum length of the certificate in bytes.
nextHigher Certificate	CryptoService Certificate	0..1	ref	The reference identifies the next higher certificate in the certificate chain. **Tags:** atp.Splitkey=nextHigherCertificate

△

Class	CryptoServiceCertificate			
serverName Identification	String	0..1	attr	Server Name Indication (SNI) is needed if the IP address hosts multiple servers (on the same port), each of them using a different certificate. If the client sends the SNI to the Server in the client hello, the server looks the SNI up in its certificate list and uses the certificate identified by the SNI.

Table C.20: CryptoServiceCertificate

Class	DataConstr			
Note	This meta-class represents the ability to specify constraints on data. **Tags:** atp.recommendedPackage=DataConstrs			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataConstrRule	DataConstrRule	*	aggr	This is one particular rule within the data constraints. **Tags:** atp.Splitkey=dataConstrRule xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=30 xml.typeElement=false xml.typeWrapperElement=false

Table C.21: DataConstr

Class	*DataInterface* (abstract)			
Note	The purpose of this meta-class is to act as an abstract base class for subclasses that share the semantics of being concerned about data (as opposed to e.g. operations).			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Subclasses	NvDataInterface, ParameterInterface, SenderReceiverInterface			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table C.22: DataInterface

Class	*DataPrototype* (abstract)			
Note	Base class for prototypical roles of any data type.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*ApplicationCompositeElementDataPrototype*, *AutosarDataPrototype*			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
swDataDef Props	SwDataDefProps	0..1	aggr	This property allows to specify data definition properties which apply on data prototype level. **Tags:** atp.Splitkey=swDataDefProps

Table C.23: DataPrototype

Class	DiagnosticEventNeeds			
Note	Specifies the abstract needs on the configuration of the Diagnostic Event Manager for one diagnostic event. Its shortName can be regarded as a symbol identifying the diagnostic event from the viewpoint of the component or module which owns this element. In case the diagnostic event specifies a production error, the shortName shall be the name of the production error.			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
deferringFid	FunctionInhibitionNeeds	*	ref	This reference contains the link to a function identifier within the FiM which is used by the monitor before delivering a result. **Tags:** atp.Splitkey=deferringFid
diagEvent Debounce Algorithm	DiagEventDebounce Algorithm	0..1	aggr	Specifies the abstract need on the Debounce Algorithm applied by the Diagnostic Event Manager. **Tags:** atp.Splitkey=diagEventDebounceAlgorithm, diag EventDebounceAlgorithm.shortName
inhibitingFid	FunctionInhibitionNeeds	0..1	ref	This represents the primary Function Inhibition Identifier used for inhibition of the diagnostic monitor. The FID might either inhibit the monitoring of a symptom or the reporting of detected faults. **Tags:** atp.Splitkey=inhibitingFid
inhibiting SecondaryFid	FunctionInhibitionNeeds	*	ref	This represents the secondary Function Inhibition Identifier used for inhibition of the diagnostic monitor. Any of the FID inhibitions leads to an inhibition of the monitoring of a symptom or the reporting of detected faults. **Tags:** atp.Splitkey=inhibitingSecondaryFid
prestored Freezeframe StoredInNvm	Boolean	0..1	attr	If the Event uses a prestored freeze-frame (using the operations PrestoreFreezeFrame and ClearPrestored FreezeFrame of the service interface DiagnosticMonitor) this attribute indicates if the Event requires the data to be stored in non-volatile memory. TRUE = Dem shall store the prestored data in non-volatile memory, FALSE = Data can be lost at shutdown (not stored in Nvm).
usesMonitor Data	Boolean	0..1	attr	This attribute defines whether additional monitor data shall be added to the reporting of events.

Table C.24: DiagnosticEventNeeds

Class	EcuInstance			
Note	ECUInstances are used to define the ECUs used in the topology. The type of the ECU is defined by a reference to an ECU specified with the ECU resource description. **Tags:** atp.recommendedPackage=EcuInstances			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	EcuInstance			
associatedComIPduGroup	ISignalIPduGroup	*	ref	With this reference it is possible to identify which ISignal IPduGroups are applicable for which Communication Connector/ ECU. Only top level ISignalIPduGroups shall be referenced by an EcuInstance. If an ISignalIPduGroup contains other ISignalIPduGroups than these contained ISignalIPdu Groups shall not be referenced by the EcuInstance. Contained ISignalIPduGroups are associated to an Ecu Instance via the top level ISignalIPduGroup. **Tags:** atp.Splitkey=associatedComIPduGroup This Attribute is only used by the AUTOSAR Classic Platform.
associated ConsumedProvided ServiceInstance Group	ConsumedProvided ServiceInstanceGroup	*	ref	With this reference it is possible to identify which ConsumedProvidedServiceInstanceGroups are applicable for which ECUInstance. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=associatedConsumedProvidedService InstanceGroup, associatedConsumedProvidedService InstanceGroup.consumedProvidedServiceInstanceGroup, associatedConsumedProvidedServiceInstance Group.variationPoint.shortLabel vh.latestBindingTime=postBuild
associatedPdurIPduGroup	PdurIPduGroup	*	ref	With this reference it is possible to identify which PduR IPdu Groups are applicable for which Communication Connector/ ECU. **Tags:** atp.Splitkey=associatedPdurIPduGroup
channel Synchronous Wakeup	Boolean	0..1	attr	If this parameter is available and set to true, then all available channels will be woken up as soon as at least one channel wakeup occurs. If PNCs are configured, then all PNCs will be requested upon a channel wakeup.
clientIdRange	ClientIdRange	0..1	aggr	Restriction of the Client Identifier for this Ecu to an allowed range of numerical values. The Client Identifier of the transaction handle is generated by the client RTE for inter-Ecu Client/Server communication. **Tags:** atp.Splitkey=clientIdRange
com Configuration GwTimeBase	TimeValue	0..1	attr	The period between successive calls to Com_Main FunctionRouteSignals of the AUTOSAR COM module in seconds. This Attribute is only used by the AUTOSAR Classic Platform.
com ConfigurationRx TimeBase	TimeValue	0..1	attr	The period between successive calls to Com_Main FunctionRx of the AUTOSAR COM module in seconds. This Attribute is only used by the AUTOSAR Classic Platform.
com ConfigurationTx TimeBase	TimeValue	0..1	attr	The period between successive calls to Com_Main FunctionTx of the AUTOSAR COM module in seconds. This Attribute is only used by the AUTOSAR Classic Platform.
comEnable MDTForCyclic Transmission	Boolean	0..1	attr	Enables for the Com module of this EcuInstance the minimum delay time monitoring for cyclic and repeated transmissions (TransmissionModeTiming has cyclic Timing assigned or eventControlledTiming with numberOf Repetitions > 0). This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	EcuInstance			
commController	Communication Controller	*	aggr	CommunicationControllers of the ECU. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=commController, commController.shortName, commController.variationPoint.shortLabel vh.latestBindingTime=postBuild
connector	Communication Connector	*	aggr	All channels controlled by a single controller. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=connector, connector.shortName, connector.variationPoint.shortLabel vh.latestBindingTime=postBuild
dltConfig	DltConfig	0..1	aggr	Describes the Dlt configuration on this EcuInstance. **Tags:** atp.Splitkey=dltConfig This Attribute is only used by the AUTOSAR Classic Platform.
doIpConfig	DoIpConfig	0..1	aggr	DoIp configuration on this EcuInstance. **Tags:** atp.Splitkey=doIpConfig atp.Status=draft This Attribute is only used by the AUTOSAR Classic Platform.
ecuTaskProxy	OsTaskProxy	*	ref	Reference to OsTaskProxies assigned to the Ecu Instance. **Tags:** atp.Splitkey=ecuTaskProxy This Attribute is only used by the AUTOSAR Classic Platform.
ethSwitchPort Group Derivation	Boolean	0..1	attr	Defines whether the derivation of SwitchPortGroups based on VLAN and/or CouplingPort.pncMapping shall be performed for this EcuInstance. If not defined the derivation shall not be done. This Attribute is only used by the AUTOSAR Classic Platform.
firewallRule	StateDependentFirewall	*	ref	Firewall rules defined in the context of an EcuInstance. **Tags:** atp.Splitkey=firewallRule
j1939Node	J1939Node	*	aggr	Optional collection of J1939Nodes defined on this Ecu Instance. **Tags:** atp.Splitkey=j1939Node, j1939Node.shortName This Attribute is only used by the AUTOSAR Classic Platform.
partition	EcuPartition	*	aggr	Optional definition of Partitions within an Ecu. **Tags:** atp.Splitkey=partition, partition.shortName This Attribute is only used by the AUTOSAR Classic Platform.
pncNmRequest	Boolean	0..1	attr	Defines if this EcuInstance shall request Nm on all its PhysicalChannels which have Nm variant set to FULL each time a PNC is requested.
pncPrepare SleepTimer	TimeValue	0..1	attr	Time in seconds the PNC state machine shall wait in PNC_PREPARE_SLEEP.
pnc Synchronous Wakeup	Boolean	0..1	attr	If this parameter is available and set to true then all available PNCs will be woken up as soon as a channel wakeup occurs. This is ensured by adding all PNCs to all channel wakeup sources during upstream mapping.

▽

△

Class	**EcuInstance**			
pnResetTime	TimeValue	0..1	attr	Specifies the runtime of the reset timer in seconds. This reset time is valid for the reset of PN requests in the EIRA and in the ERA.
sleepMode Supported	Boolean	0..1	attr	Specifies whether the ECU instance may be put to a "low power mode" • true: sleep mode is supported • false: sleep mode is not supported Note: This flag may only be set to "true" if the feature is supported by both hardware and basic software. This Attribute is only used by the AUTOSAR Classic Platform.
tcpIpIcmpProps	EthTcpIpIcmpProps	0..1	ref	EcuInstance specific ICMP (Internet Control Message Protocol) attributes **Tags:** atp.Splitkey=tcpIpIcmpProps This Attribute is only used by the AUTOSAR Classic Platform.
tcpIpProps	EthTcpIpProps	0..1	ref	EcuInstance specific TcpIp Stack attributes. **Tags:** atp.Splitkey=tcpIpProps This Attribute is only used by the AUTOSAR Classic Platform.
wakeUpOver BusSupported	Boolean	0..1	attr	Driver support for wakeup over Bus. This Attribute is only used by the AUTOSAR Classic Platform.

Table C.25: EcuInstance

Class	**EthernetPhysicalChannel**			
Note	The EthernetPhysicalChannel represents a VLAN or an untagged channel. An untagged channel is modeled as an EthernetPhysicalChannel without an aggregated VLAN.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
comMLight Timeout	TimeValue	0..1	attr	Defines the timeout (in seconds) after COMM_FULL_COMMUNICATION sub-state COMM_FULL_COM_READY_SLEEP is left in the ComM channel state machine.
network Endpoint	NetworkEndpoint	*	aggr	Collection of NetworkEndpoints that are used in the VLan. **Tags:** atp.Splitkey=networkEndpoint, network Endpoint.shortName
soAdConfig	SoAdConfig	0..1	aggr	SoAd Configuration for one specific Physical Channel. **Tags:** atp.Splitkey=soAdConfig This Attribute is only used by the AUTOSAR Classic Platform.
vlan	VlanConfig	0..1	aggr	VLAN Configuration. **Tags:** atp.Splitkey=vlan, vlan.shortName

Table C.26: EthernetPhysicalChannel

Class	EvaluatedVariantSet			
Note	This meta class represents the ability to express if a set of ARElements is able to support one or more particular variants. In other words, for a given set of evaluatedElements this meta class represents a table of evaluated variants, where each PredefinedVariant represents one column. In this column each descendant sw SystemconstantValue resp. postbuildVariantCriterionValue represents one entry. In a graphical representation each swSystemconstantValueSet / postBuildVariantCriterionValueSet could be used as an intermediate headline in the table column. If the approvalStatus is "APPROVED" it expresses that the collection of CollectableElements is known be valid for the given evaluatedVariants. Note that the EvaluatedVariantSet is a CollectableElement. This allows to establish a hierarchy of EvaluatedVariantSets. **Tags:** atp.recommendedPackage=EvaluatedVariantSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
approvalStatus	NameToken	0..1	attr	Defines the approval status of a predefined variant. Two values are predefined: "APPROVED" and "REJECTED": • Approved variants are known to work. • Rejected variants are known NOT to work. Further values can be approved on a per-company basis; within AUTOSAR only "APPROVED" and "REJECTED" should be recognized.
evaluated Element	CollectableElement	*	ref	This represents a particular element which is evaluated in context of the EvaluatedVariants. The approvalStatus applies to this element (and all of its descendants). In other words, the referenced elements are those that were considered when the predefined variant was evaluated. **Tags:** atp.Splitkey=evaluatedElement
evaluated Variant	PredefinedVariant	*	ref	This metaclass represents one particular variant which was evaluated. LowerMultiplicity is set to 0 to support a stepwise approach. **Tags:** atp.Splitkey=evaluatedVariant

Table C.27: EvaluatedVariantSet

Class	ISignal			
Note	Signal of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal is sent in different SignalIPdus to multiple receivers. To support the RTE "signal fan-out" each SignalIPdu contains ISignals. If the same System Signal is to be mapped into several SignalIPdus there is one ISignal needed for each ISignalToIPduMapping. ISignals describe the Interface between the Precompile configured RTE and the potentially Postbuild configured Com Stack (see ECUC Parameter Mapping). In case of the SystemSignalGroup an ISignal shall be created for each SystemSignal contained in the SystemSignalGroup. **Tags:** atp.recommendedPackage=ISignals			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	ISignal			
data Transformation	DataTransformation	0..1	ref	Optional reference to a DataTransformation which represents the transformer chain that is used to transform the data that shall be placed inside this ISignal. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataTransformation, data Transformation.dataTransformation, data Transformation.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime
dataTypePolicy	DataTypePolicyEnum	0..1	attr	With the aggregation of SwDataDefProps an ISignal specifies how it is represented on the network. This representation follows a particular policy. Note that this causes some redundancy which is intended and can be used to support flexible development methodology as well as subsequent integrity checks. If the policy "networkRepresentationFromComSpec" is chosen the network representation from the ComSpec that is aggregated by the PortPrototype shall be used. If the "override" policy is chosen the requirements specified in the PortInterface and in the ComSpec are not fulfilled by the networkRepresentationProps. In case the System Description doesn't use a complete Software Component Description (VFB View) the "legacy" policy can be chosen.
initValue	ValueSpecification	0..1	aggr	Optional definition of a ISignal's initValue in case the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy system signals. This value can be used to configure the Signal's "Init Value". If a full DataMapping exist for the SystemSignal this information may be available from a configured Sender ComSpec and ReceiverComSpec. In this case the initvalues in SenderComSpec and/or ReceiverComSpec override this optional value specification. Further restrictions apply from the RTE specification. **Tags:** atp.Splitkey=initValue
iSignalProps	ISignalProps	0..1	aggr	Additional optional ISignal properties that may be stored in different files. **Tags:** atp.Splitkey=iSignalProps
iSignalType	ISignalTypeEnum	0..1	attr	This attribute defines whether this iSignal is an array that results in a UINT8_N / UINT8_DYN ComSignalType in the COM configuration or a primitive type.
length	UnlimitedInteger	0..1	attr	Size of the signal in bits. The size needs to be derived from the mapped VariableDataPrototype according to the mapping of primitive DataTypes to BaseTypes as used in the RTE. Indicates maximum size for dynamic length signals. The ISignal length of zero bits is allowed.

▽

△

Class	**ISignal**			
network Representation Props	SwDataDefProps	0..1	aggr	Specification of the actual network representation. The usage of SwDataDefProps for this purpose is restricted to the attributes compuMethod and baseType. The optional baseType attributes "memAllignment" and "byteOrder" shall not be used. The attribute "dataTypePolicy" in the SystemTemplate element defines whether this network representation shall be ignored and the information shall be taken over from the network representation of the ComSpec. If "override" is chosen by the system integrator the network representation can violate against the requirements defined in the PortInterface and in the network representation of the ComSpec. In case that the System Description doesn't use a complete Software Component Description (VFB View) this element is used to configure "ComSignalDataInvalid Value" and the Data Semantics. **Tags:** atp.Splitkey=networkRepresentationProps
systemSignal	SystemSignal	0..1	ref	Reference to the System Signal that is supposed to be transmitted in the ISignal. **Tags:** atp.Splitkey=systemSignal
timeout Substitution Value	ValueSpecification	0..1	aggr	Defines and enables the ComTimeoutSubstituition for this ISignal. **Tags:** atp.Splitkey=timeoutSubstitutionValue
transformation ISignalProps	TransformationISignal Props	*	aggr	A transformer chain consists of an ordered list of transformers. The ISignal specific configuration properties for each transformer are defined in the TransformationISignalProps class. The transformer configuration properties that are common for all ISignals are described in the TransformationTechnology class. **Tags:** atp.Splitkey=transformationISignalProps

Table C.28: ISignal

Class	**ISignalGroup**			
Note	SignalGroup of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal Group is sent in different SignalIPdus to multiple receivers. An ISignalGroup refers to a set of ISignals that shall always be kept together. A ISignalGroup represents a COM Signal Group. Therefore it is recommended to put the ISignalGroup in the same Package as ISignals (see atp.recommendedPackage) **Tags:** atp.recommendedPackage=ISignalGroups			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**ISignalGroup**			
comBased SignalGroup Transformation	DataTransformation	0..1	ref	Optional reference to a DataTransformation which represents the transformer chain that is used to transform the data that shall be placed inside this ISignalGroup based on the COMBasedTransformer approach. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=comBasedSignalGroupTransformation, comBasedSignalGroupTransformation.dataTransformation, comBasedSignalGroupTransformation.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime
iSignal	ISignal	*	ref	Reference to a set of ISignals that shall always be kept together. **Tags:** atp.Splitkey=iSignal
systemSignal Group	SystemSignalGroup	0..1	ref	Reference to the SystemSignalGroup that is defined on VFB level and that is supposed to be transmitted in the ISignalGroup. **Tags:** atp.Splitkey=systemSignalGroup
transformation ISignalProps	TransformationISignal Props	*	aggr	A transformer chain consists of an ordered list of transformers. The ISignalGroup specific configuration properties for each transformer are defined in the TransformationISignalProps class. The transformer configuration properties that are common for all ISignal Groups are described in the TransformationTechnology class. **Tags:** atp.Splitkey=transformationISignalProps

Table C.29: ISignalGroup

Class	**ISignalIPduGroup**			
Note	The AUTOSAR COM Layer is able to start and to stop sending and receiving configurable groups of I-Pdus during runtime. An ISignalIPduGroup contains either ISignalIPdus or ISignalIPduGroups. **Tags:** atp.recommendedPackage=ISignalIPduGroups			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Direction	Communication DirectionType	0..1	attr	This attribute determines in which direction IPdus that are contained in this IPduGroup will be transmitted (communication direction can be either In or Out).
communication Mode	String	0..1	attr	This attribute defines the use-case for this ISignalIPdu Group (e.g. diagnostic, debugging etc.). For example, in a diagnostic mode all IPdus - which are not involved in diagnostic - are disabled. The use cases are not limited to a fixed enumeration and can be specified as a string.
contained ISignalIPdu Group	ISignalIPduGroup	*	ref	An I-Pdu group can be included in other I-Pdu groups. Contained I-Pdu groups shall not be referenced by the EcuInstance. **Tags:** atp.Splitkey=containedISignalIPduGroup

▽

△

Class	**ISignalIPduGroup**			
iSignalIPdu	ISignalIPdu	*	ref	Reference to a set of Signal I-Pdus, which are contained in the ISignal I-Pdu Group. atpVariation: The content of a ISignal I-Pdu group can vary (->vehicle modes). **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iSignalIPdu, iSignalIPdu.iSignalIPdu, iSignalIPdu.variationPoint.shortLabel vh.latestBindingTime=postBuild
nmPdu	NmPdu	*	ref	Reference to a set of NmPdus with NmUserData, which are contained in the ISignalIPduGroup. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nmPdu, nmPdu.nmPdu, nmPdu.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table C.30: ISignalIPduGroup

Class	**ISignalPort**			
Note	Connectors reception or send port on the referenced channel referenced by an ISignalTriggering. If different timeouts or DataFilters for ISignals need to be specified several ISignalPorts may be created.			
Base	*ARObject*, *CommConnectorPort*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*CommunicationConnector*.ecuCommPortInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataFilter	DataFilter	0..1	aggr	Optional specification of a signal COM filter at the receiver side in case that the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy system signals. If a full DataMapping exist for the SystemSignal this information may be available from a configured ReceiverComSpec. In this case the ReceiverComSpec overrides this optional specification. **Tags:** atp.Splitkey=dataFilter
ddsQosProfile	DdsCpQosProfile	0..1	ref	Reference to the DDS Qos profile used for this ISignal. **Tags:** atp.Splitkey=ddsQosProfile atp.Status=candidate
firstTimeout	TimeValue	0..1	attr	• ISignalPort with communicationDirection = in: Optional first timeout value in seconds for the reception of the ISignal. • ISignalPort with communicationDirection = out: Optional first timeout value in seconds for transmission deadline monitoring.
handleInvalid	HandleInvalidEnum	0..1	attr	This attribute defines how invalidation is applied to the ISignals received in the context of this ISignalPort.

▽

△

Class	**ISignalPort**			
timeout	TimeValue	0..1	attr	• ISignalPort with communicationDirection = in: Optional timeout value in seconds for the reception of the ISignal. The attribute value is used to configure the Com Timeout in the COM module. The RTE ignores this attribute. The timeout can also be specified with the NonqueuedReceiverComSpec.aliveTimeout attribute. If a full DataMapping exists for the SystemSignal and the value is available in the configured ReceiverComSpec, then the timeout value in the ReceiverComSpec overrides this optional timeout specification during the creation of the Base Ecu Configuration of the COM module. • ISignalPort with communicationDirection = out: Optional timeout value in seconds for the transmission of the ISignal. The attribute value is used to configure the ComTimeout in the COM module. The RTE ignores this attribute. The timeout can also be specified with the SenderComSpec.transmissionAcknowledge.timeout attribute. If a full DataMapping exists for the System Signal and the value is available in the configured Sender ComSpec, then the timeout value in the SenderComSpec overrides this optional timeout specification during the creation of the Base Ecu Configuration of the COM module. This attribute can be used in the following cases: • legacy signal where the System Description doesn't use a complete Software Component Description (VFB View) and where the DataMapping is missing. • bus monitoring use cases in which the DataMapping is ignored.

Table C.31: ISignalPort

Class	**ISignalTriggering**			
Note	A ISignalTriggering allows an assignment of ISignals to physical channels.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*PhysicalChannel*.iSignalTriggering			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
iSignal	ISignal	0..1	ref	This reference shall be used if an ISignal is transported on the PhysicalChannel. This reference forms an XOR relationship with the ISignalTriggering-ISignalGroup reference. **Tags:** atp.Splitkey=iSignal
iSignalGroup	ISignalGroup	0..1	ref	This reference shall be used if an ISignalGroup is transported on the PhysicalChannel. This reference forms an XOR relationship with the ISignal Triggering-ISignal reference. **Tags:** atp.Splitkey=iSignalGroup
iSignalPort	ISignalPort	*	ref	References to the ISignalPort on every ECU of the system which sends and/or receives the ISignal. References for both the sender and the receiver side shall be included when the system is completely defined. **Tags:** atp.Splitkey=iSignalPort

Table C.32: ISignalTriggering

Class	***IdentCaption*** (abstract)			
Note	This meta-class represents the caption. This allows having some meta-classes optionally identifiable.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	BswServiceDependencyIdent, DiagnosticParameterIdent, ExternalTriggeringPointIdent, ModeAccess PointIdent			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table C.33: IdentCaption

Class	***Identifiable*** (abstract)
Note	Instances of this class can be referred to by their identifier (within the namespace borders). In addition to this, Identifiables are objects which contribute significantly to the overall structure of an AUTOSAR description. In particular, Identifiables might contain Identifiables.
Base	*ARObject*, *MultilanguageReferrable*, *Referrable*
Subclasses	ARPackage, *AbstractDoIpLogicAddressProps*, *AbstractEvent*, *AbstractImplementationDataTypeElement*, *AbstractSecurityEventFilter*, *AbstractSecurityIdsmInstanceFilter*, *AbstractServiceInstance*, AppOsTask ProxyToEcuTaskProxyMapping, ApplicationEndpoint, ApplicationError, ApplicationPartitionToEcuPartition Mapping, AppliedStandard, AsynchronousServerCallResultPoint, *AtpBlueprint*, *AtpClassifier*, *Atp Feature*, AutosarOperationArgumentInstance, AutosarVariableInstance, *BinaryManifestAddressable Object*, BinaryManifestItemDefinition, *BinaryManifestResource*, BinaryManifestResourceDefinition, Block State, BswInternalTriggeringPoint, BswModuleDependency, *BuildActionEntity*, BuildActionEnvironment, CanSecSecureChannel, CanTpAddress, CanTpChannel, CanTpNode, Chapter, ClientIdDefinition, Client ServerOperation, Code, *CollectableElement*, ComManagementMapping, *CommConnectorPort*, *CommunicationConnector*, *CommunicationController*, Compiler, ConsistencyNeeds, ConsumedEvent Group, *CouplingElementAbstractDetails*, CouplingPort, *CouplingPortAbstractShaper*, *CouplingPort StructuralElement*, *CpSoftwareClusterResource*, CpSoftwareClusterResourceToApplicationPartition Mapping, CpSoftwareClusterToApplicationPartitionMapping, CpSoftwareClusterToEcuInstanceMapping, CpSoftwareClusterToResourceMapping, *CryptoServiceMapping*, CyclicHandlingComDataToOsTask ProxyMapping, DataPrototypeGroup, DataPrototypeTransformationPropsIdent, DataTransformation, *Dds AbstractServiceInstanceElementCp*, DdsCpDomain, DdsCpPartition, DdsCpQosProfile, DdsCpTopic, DependencyOnArtifact, *DiagEventDebounceAlgorithm*, DiagnosticAuthTransmitCertificateEvaluation, DiagnosticConnectedIndicator, DiagnosticDataElement, DiagnosticDebounceAlgorithmProps, Diagnostic ExtendedDataRecordElement, DiagnosticFunctionInhibitSource, DiagnosticParameterElement, *DiagnosticRoutineSubfunction*, DltApplication, DltArgument, DltArgumentProps, DltLogChannel, Dlt Message, DoIpInterface, DoIpLogicAddress, DoIpRoutingActivation, ECUMapping, *EOCExecutableEntity RefAbstract*, EcuPartition, EcuPartitionToCoreMapping, EcucContainerValue, *EcucDefinitionElement*, EcucDestinationUriDef, EcucEnumerationLiteralDef, EcucQuery, EcucValidationCondition, Ethernet WakeupSleepOnDatalineConfig, EventHandler, ExclusiveArea, *ExecutableEntity*, *ExecutionTime*, FMAttributeDef, FMFeatureMapAssertion, FMFeatureMapCondition, FMFeatureMapElement, FMFeature Relation, FMFeatureRestriction, FMFeatureSelection, FlatInstanceDescriptor, FlexrayTpConnection Control, FlexrayTpNode, FlexrayTpPduPool, *FrameTriggering*, GlobalTimeGateway, *GlobalTimeMaster*, *GlobalTimeSlave*, *HeapUsage*, HwAttributeDef, HwAttributeLiteralDef, HwPin, HwPinGroup, *IEEE1722 TpAcfBus*, *IEEE1722TpAcfBusPart*, IPSecRule, IPv6ExtHeaderFilterList, ISignalToIPduMapping, ISignal Triggering, *IdentCaption*, ImpositionTime, InternalTriggeringPoint, J1939Node, J1939SharedAddress Cluster, J1939TpNode, Keyword, LifeCycleState, LinScheduleTable, LinTpNode, Linker, MacAddress VlanMembership, MacMulticastGroup, MacSecKayParticipant, McDataInstance, MemorySection, Mode Declaration, ModeDeclarationMapping, ModeSwitchPoint, ModeSwitchSenderComSpecProps, Network Endpoint, *NmCluster*, NmEcu, *NmNode*, NvBlockDescriptor, *PackageableElement*, ParameterAccess, PduActivationRoutingGroup, PduToFrameMapping, PduTriggering, PerInstanceMemory, *Physical Channel*, PortElementToCommunicationResourceMapping, PortGroup, *PortInterfaceMapping*, Queued ReceiverComSpecProps, ResourceConsumption, RootSwCompositionPrototype, RptComponent, Rpt Container, RptExecutableEntity, RptExecutableEntityEvent, RptExecutionContext, RptProfile, RptService Point, RteEventInCompositionSeparation, RteEventInCompositionToOsTaskProxyMapping, RteEventIn SystemSeparation, RteEventInSystemToOsTaskProxyMapping, RunnableEntityGroup, *SdgAttribute*, Sdg Class, SecOcJobRequirement, SecureCommunicationAuthenticationProps, SecureCommunication FreshnessProps, SecurityEventContextDataElement, SecurityEventContextProps, *ServerCallPoint*, ServerComSpecProps, *ServiceNeeds*, SignalServiceTranslationElementProps, SignalServiceTranslation EventProps, SignalServiceTranslationProps, SocketAddress, SomeipTpChannel, *StackUsage*, Static ▽

▽

△

Class	***Identifiable*** (abstract)			
	△ SocketConnection, StructuredReq, SwGenericAxisParamType, SwServiceArg, SwcServiceDependency, SwcToApplicationPartitionMapping, SwcToEcuMapping, SwcToImplMapping, SwitchAsynchronousTrafficShaperGroupEntry, SwitchAtsInstanceEntry, SwitchFlowMeteringEntry, SwitchStreamFilterActionDestPortModification, SwitchStreamFilterEntry, SwitchStreamFilterRule, SwitchStreamGateEntry, SwitchStreamIdentification, SystemMapping, SystemSignalGroupToCommunicationResourceMapping, SystemSignalToCommunicationResourceMapping, TDCpSoftwareClusterMapping, TDCpSoftwareClusterResourceMapping, TcpOptionFilterList, *TimingClock*, TimingClockSyncAccuracy, TimingCondition, *TimingConstraint*, *TimingDescription*, TimingExtensionResource, TimingModeInstance, TlsCryptoCipherSuite, TlsCryptoCipherSuiteProps, Topic1, TpAddress, TraceableTable, TraceableText, *TracedFailure*, TransformationISignalPropsIdent, *TransformationProps*, TransformationTechnology, Trigger, VariableAccess, VariationPointProxy, ViewMap, VlanConfig, WaitPoint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
adminData	AdminData	0..1	aggr	This represents the administrative data for the identifiable object. **Tags:** atp.Splitkey=adminData xml.sequenceOffset=-40
annotation	Annotation	*	aggr	Possibility to provide additional notes while defining a model element (e.g. the ECU Configuration Parameter Values). These are not intended as documentation but are mere design notes. **Tags:** atp.Splitkey=annotation xml.sequenceOffset=-25
category	CategoryString	0..1	attr	The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints. **Tags:** xml.sequenceOffset=-50
desc	MultiLanguageOverviewParagraph	0..1	aggr	This represents a general but brief (one paragraph) description what the object in question is about. It is only one paragraph! Desc is intended to be collected into overview tables. This property helps a human reader to identify the object in question. More elaborate documentation, (in particular how the object is built or used) should go to "introduction". **Tags:** atp.Splitkey=desc xml.sequenceOffset=-60
introduction	DocumentationBlock	0..1	aggr	This represents more information about how the object in question is built or is used. Therefore it is a DocumentationBlock. **Tags:** atp.Splitkey=introduction xml.sequenceOffset=-30

▽

△

Class	***Identifiable*** (abstract)			
uuid	String	0..1	attr	The purpose of this attribute is to provide a globally unique identifier for an instance of a meta-class. The values of this attribute should be globally unique strings prefixed by the type of identifier. For example, to include a DCE UUID as defined by The Open Group, the UUID would be preceded by "DCE:". The values of this attribute may be used to support merging of different AUTOSAR models. The form of the UUID (Universally Unique Identifier) is taken from a standard defined by the Open Group (was Open Software Foundation). This standard is widely used, including by Microsoft for COM (GUIDs) and by many companies for DCE, which is based on CORBA. The method for generating these 128-bit IDs is published in the standard and the effectiveness and uniqueness of the IDs is not in practice disputed. If the id namespace is omitted, DCE is assumed. An example is "DCE:2fac1234-31f8-11b4-a222-08002b34c003". The uuid attribute has no semantic meaning for an AUTOSAR model and there is no requirement for AUTOSAR tools to manage the timestamp. **Tags:** xml.attribute=true

Table C.34: Identifiable

Class	***InternalBehavior*** (abstract)			
Note	Common base class (abstract) for the internal behavior of both software components and basic software modules/clusters.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	BswInternalBehavior, SwcInternalBehavior			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
constant Memory	ParameterData Prototype	*	aggr	Describes a read only memory object containing characteristic value(s) implemented by this Internal Behavior. The shortName of ParameterDataPrototype has to be equal to the ''C' identifier of the described constant. The characteristic value(s) might be shared between Sw ComponentPrototypes of the same SwComponentType. The aggregation of constantMemory is subject to variability with the purpose to support variability in the software component or module implementations. Typically different algorithms in the implementation are requiring different number of memory objects. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=constantMemory, constantMemory.short Name, constantMemory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
constantValue Mapping	ConstantSpecification MappingSet	*	ref	Reference to the ConstantSpecificationMapping to be applied for the particular InternalBehavior **Tags:** atp.Splitkey=constantValueMapping
dataType Mapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMapping to be applied for the particular InternalBehavior **Tags:** atp.Splitkey=dataTypeMapping

▽

△

Class	***InternalBehavior*** (abstract)			
exclusiveArea	ExclusiveArea	*	aggr	This specifies an ExclusiveArea for this InternalBehavior. The exclusiveArea is local to the component resp. module. The aggregation of ExclusiveAreas is subject to variability. Note: the number of ExclusiveAreas might vary due to the conditional existence of RunnableEntities or BswModuleEntities. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=exclusiveArea, exclusiveArea.shortName, exclusiveArea.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
exclusiveArea NestingOrder	ExclusiveAreaNesting Order	*	aggr	This represents the set of ExclusiveAreaNestingOrder owned by the InternalBehavior. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=exclusiveAreaNestingOrder, exclusiveArea NestingOrder.shortName, exclusiveAreaNesting Order.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
staticMemory	VariableDataPrototype	*	aggr	Describes a read and writeable static memory object representing measurerment variables implemented by this software component. The term "static" is used in the meaning of "non-temporary" and does not necessarily specify a linker encapsulation. This kind of memory is only supported if supportsMultipleInstantiation is FALSE. The shortName of the VariableDataPrototype has to be equal with the ''C' identifier of the described variable. The aggregation of staticMemory is subject to variability with the purpose to support variability in the software component's implementations. Typically different algorithms in the implementation are requiring different number of memory objects. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=staticMemory, staticMemory.shortName, staticMemory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table C.35: InternalBehavior

Class	**J1939DcmIPdu**			
Note	Represents the IPdus handled by J1939Dcm. **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnostic MessageType	PositiveInteger	0..1	attr	This attribute is used to identify the actual DMx message, e.g 1 means DM01, etc.

Table C.36: J1939DcmIPdu

Class	J1939NmNode			
Note	J1939 specific NM Node attributes. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmNode*, *Referrable*			
Aggregated by	*NmCluster*.nmNode			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
address Configuration Capability	J1939NmAddress ConfigurationCapability Enum	0..1	attr	Defines the Address Configuration Capability of the J1939NmNode (corresponding to an SAE J1939 Controller Application, CA).
j1939Node	J1939Node	0..1	ref	Reference to J1939Node that has a defined J1939Node Name and is assocated with a J1939Controller Application. **Tags:** atp.Splitkey=j1939Node

Table C.37: J1939NmNode

Class	ModeDeclaration			
Note	Declaration of one Mode. The name and semantics of a specific mode is not defined in the meta-model.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, ModeDeclarationGroup.modeDeclaration			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
value	PositiveInteger	0..1	attr	The RTE shall take the value of this attribute for generating the source code representation of this Mode Declaration.

Table C.38: ModeDeclaration

Class	ModeDeclarationGroup			
Note	A collection of Mode Declarations. Also, the initial mode is explicitly identified. **Tags:** atp.recommendedPackage=ModeDeclarationGroups			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *Uploadable PackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
initialMode	ModeDeclaration	0..1	ref	The initial mode of the ModeDeclarationGroup. This mode is active before any mode switches occurred. **Tags:** atp.Splitkey=initialMode
mode Declaration	ModeDeclaration	*	aggr	The ModeDeclarations collected in this ModeDeclaration Group. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=modeDeclaration, modeDeclaration.short Name, modeDeclaration.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime
modeManager ErrorBehavior	ModeErrorBehavior	0..1	aggr	This represents the ability to define the error behavior expected by the mode manager in case of errors on the mode user side (e.g. terminated mode user). **Tags:** atp.Splitkey=modeManagerErrorBehavior This Attribute is only used by the AUTOSAR Classic Platform.



△

Class	ModeDeclarationGroup			
modeTransition	ModeTransition	*	aggr	This represents the avaliable ModeTransitions of the ModeDeclarationGroup **Tags:** atp.Splitkey=modeTransition, modeTransition.shortName This Attribute is only used by the AUTOSAR Classic Platform.
modeUserErrorBehavior	ModeErrorBehavior	0..1	aggr	This represents the definition of the error behavior expected by the mode user in case of errors on the mode manager side (e.g. terminated mode manager). **Tags:** atp.Splitkey=modeUserErrorBehavior This Attribute is only used by the AUTOSAR Classic Platform.
onTransitionValue	PositiveInteger	0..1	attr	The value of this attribute shall be taken into account by the RTE generator for programmatically representing a value used for the transition between two statuses. This Attribute is only used by the AUTOSAR Classic Platform.

Table C.39: ModeDeclarationGroup

Class	ModeDeclarationGroupPrototype			
Note	The ModeDeclarationGroupPrototype specifies a set of Modes (ModeDeclarationGroup) which is provided or required in the given context.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, BswModuleDescription.providedModeGroup, BswModuleDescription.requiredModeGroup, FirewallStateSwitchInterface.firewallStateMachine, FunctionGroupSet.functionGroup, ModeSwitchInterface.modeGroup, Process.processStateMachine, StateManagementStateNotification.stateMachine			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
swCalibrationAccess	SwCalibrationAccessEnum	0..1	attr	This allows for specifying whether or not the enclosing ModeDeclarationGroupPrototype can be measured at run-time. This Attribute is only used by the AUTOSAR Classic Platform.
type	ModeDeclarationGroup	0..1	tref	The "collection of ModeDeclarations" (= ModeDeclarationGroup) supported by a component **Stereotypes:** isOfType **Tags:** atp.Splitkey=type

Table C.40: ModeDeclarationGroupPrototype

Class	ModeInBswModuleDescriptionInstanceRef			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	*BswEvent*.disabledInMode, BswModeSwitchEvent.mode, DiagnosticEnvBswModeElement.mode			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
base	BswModuleDescription	0..1	ref	**Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10 This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	ModeInBswModuleDescriptionInstanceRef			
contextMode Declaration Group	ModeDeclarationGroup Prototype	0..1	ref	**Tags:** xml.sequenceOffset=20
targetMode	ModeDeclaration	0..1	ref	**Tags:** xml.sequenceOffset=30

Table C.41: ModeInBswModuleDescriptionInstanceRef

Primitive	NameToken
Note	This is an identifier as used in xml, e.g. xml-names. Typical usages are, for example, the names of type emitters, protocols, or profiles. For details see NMTOKEN definition on the W3C website (https://www.w3.org/TR/xml/#NT-Nmtoken). Note: Although NameToken supports a wide range of characters, the actually allowed patterns for a certain attribute typed by NameToken may be further restricted by the specification of that attribute. **Tags:** xml.xsd.customType=NMTOKEN-STRING xml.xsd.type=NMTOKEN

Table C.42: NameToken

Class	NvDataInterface			
Note	A non volatile data interface declares a number of `VariableDataPrototype`s to be exchanged between non volatile block components and atomic software components. **Tags:** atp.recommendedPackage=PortInterfaces			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *DataInterface*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
nvData	VariableDataPrototype	*	aggr	The VariableDataPrototype of this nv data interface. **Tags:** atp.Splitkey=nvData, nvData.shortName

Table C.43: NvDataInterface

Class	ObdMonitorServiceNeeds			
Note	Specifies the abstract needs of a component or module on the configuration of OBD Services in relation to a particular on-board monitoring test supported by this component or module. (OBD Service 06).			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
applicationData Type	ApplicationDataType	0..1	ref	reference to an ApplicationDataType that describes the scaling of the data reported by the software-component to the Dem. **Tags:** atp.Splitkey=applicationDataType
eventNeeds	DiagnosticEventNeeds	0..1	ref	This reference identifies the corresponding diagnostic event. **Tags:** atp.Splitkey=eventNeeds
unitAndScaling Id	PositiveInteger	0..1	attr	Unit and scaling ID according to ISO 15031-5.
updateKind	DiagnosticMonitor UpdateKindEnum	0..1	attr	This attribute indicates the settings for the acceptance of updates to the Dem.

Table C.44: ObdMonitorServiceNeeds

Class	PPortPrototype			
Note	Component port providing a certain port interface.			
Base	*ARObject*, *AbstractProvidedPortPrototype*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *PortPrototype*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
provided Interface	PortInterface	0..1	tref	The interface that this port provides. **Stereotypes:** isOfType **Tags:** atp.Splitkey=providedInterface

Table C.45: PPortPrototype

Class	***PhysicalChannel*** (abstract)			
Note	A physical channel is the transmission medium that is used to send and receive information between communicating ECUs. Each CommunicationCluster has at least one physical channel. Bus systems like CAN and LIN only have exactly one PhysicalChannel. A FlexRay cluster may have more than one PhysicalChannels that may be used in parallel for redundant communication. An ECU is part of a cluster if it contains at least one controller that is connected to at least one channel of the cluster.#			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AbstractCanPhysicalChannel*, EthernetPhysicalChannel, FlexrayPhysicalChannel, LinPhysicalChannel, UserDefinedPhysicalChannel			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
comm Connector	Communication Connector	*	ref	Reference to the ECUInstance via a Communication Connector to which the channel is connected. atpVariation: Variable assignment of Physical Channels to different CommunicationConnectors is expressed with this variation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=commConnector, comm Connector.communicationConnector, comm Connector.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
frameTriggering	FrameTriggering	*	aggr	One frame triggering is defined for exactly one channel. Channels may have assigned an arbitrary number of frame triggerings. atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=frameTriggering, frameTriggering.shortName, frameTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.



△

Class	***PhysicalChannel*** (abstract)			
iSignal Triggering	ISignalTriggering	*	aggr	One ISignalTriggering is defined for exactly one channel. Channels may have assigned an arbitrary number of ISignaltriggerings. atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iSignalTriggering, iSignalTriggering.shortName, iSignalTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
managed Physical Channel	PhysicalChannel	*	ref	Reference between a channel with role managing channel and a channel with role managed channel. **Tags:** atp.Splitkey=managedPhysicalChannel This Attribute is only used by the AUTOSAR Classic Platform.
pduTriggering	PduTriggering	*	aggr	One PduTriggering is defined for exactly one channel. Channels may have assigned an arbitrary number of I-Pdu triggerings. atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=pduTriggering, pduTriggering.shortName, pduTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.

Table C.46: PhysicalChannel

Class	***PortInterface*** (abstract)			
Note	Abstract base class for an interface that is either provided or required by a port of a software component.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	ClientServerInterface, *DataInterface*, ModeSwitchInterface, TriggerInterface			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	***PortInterface*** (abstract)			
isService	Boolean	0..1	attr	This flag is set if the `PortInterface` is to be used for communication between an • `ApplicationSwComponentType` or • `ServiceProxySwComponentType` or • `SensorActuatorSwComponentType` or • `ComplexDeviceDriverSwComponentType` • `ServiceSwComponentType` • `EcuAbstractionSwComponentType` and a `ServiceSwComponentType` (namely an AUTOSAR Service) located on the same ECU. Otherwise the flag is not set. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=blueprintDerivationTime This Attribute is only used by the AUTOSAR Classic Platform.
serviceKind	ServiceProviderEnum	0..1	attr	This attribute provides further details about the nature of the applied service. This Attribute is only used by the AUTOSAR Classic Platform.

Table C.47: PortInterface

Class	***PortPrototype*** (abstract)			
Note	Base class for the ports of an AUTOSAR software component. The aggregation of PortPrototypes is subject to variability with the purpose to support the conditional existence of ports.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AbstractProvidedPortPrototype*, *AbstractRequiredPortPrototype*			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
clientServer Annotation	ClientServerAnnotation	*	aggr	Annotation of this `PortPrototype` with respect to client/server communication. **Tags:** atp.Splitkey=clientServerAnnotation
delegatedPort Annotation	DelegatedPort Annotation	0..1	aggr	Annotations on this delegated port. **Tags:** atp.Splitkey=delegatedPortAnnotation
ioHwAbstraction Server Annotation	IoHwAbstractionServer Annotation	*	aggr	Annotations on this IO Hardware Abstraction port. **Tags:** atp.Splitkey=ioHwAbstractionServerAnnotation
modePort Annotation	ModePortAnnotation	*	aggr	Annotations on this mode port. **Tags:** atp.Splitkey=modePortAnnotation
nvDataPort Annotation	NvDataPortAnnotation	*	aggr	Annotations on this non voilatile data port. **Tags:** atp.Splitkey=nvDataPortAnnotation
parameterPort Annotation	ParameterPort Annotation	*	aggr	Annotations on this parameter port. **Tags:** atp.Splitkey=parameterPortAnnotation
senderReceiver Annotation	SenderReceiver Annotation	*	aggr	Collection of annotations of this ports sender/receiver communication. **Tags:** atp.Splitkey=senderReceiverAnnotation
triggerPort Annotation	TriggerPortAnnotation	*	aggr	Annotations on this trigger port. **Tags:** atp.Splitkey=triggerPortAnnotation

Table C.48: PortPrototype

Class	PostBuildVariantCriterion			
Note	This class specifies one particular PostBuildVariantSelector. **Tags:** atp.recommendedPackage=PostBuildVariantCriterions			
Base	*ARElement*, *ARObject*, *AtpDefinition*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
compuMethod	CompuMethod	0..1	ref	The compuMethod specifies the possible values for the variant criterion serving as an enumerator. **Tags:** atp.Splitkey=compuMethod

Table C.49: PostBuildVariantCriterion

Primitive	Ref			
Note	This primitive denotes a name based reference. For detailed syntax see the xsd.pattern. • first slash (relative or absolute reference) [optional] • Identifier [required] • a sequence of slashes and Identifiers [optional] This primitive is used by the meta-model tools to create the references. **Tags:** xml.xsd.customType=REF xml.xsd.pattern=/?[a-zA-Z][a-zA-Z0-9_]{0,127}(/[a-zA-Z][a-zA-Z0-9_]{0,127})* xml.xsd.type=string			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
base	Identifier	0..1	attr	This attribute reflects the base to be used for this reference. **Tags:** xml.attribute=true
blueprintValue	String	0..1	attr	This represents a description that documents how the value shall be defined when deriving objects from the blueprint. **Tags:** atp.Status=valid xml.attribute=true
index	PositiveInteger	0..1	attr	This attribute supports the use case to point on specific elements in an array. This is in particular required if arrays are used to implement particular data objects. The counting of array indices starts with the value 0, i.e. the index of the first array element is 0. **Tags:** xml.attribute=true

Table C.50: Ref

Class	**Referrable** (abstract)			
Note	Instances of this class can be referred to by their identifier (while adhering to namespace borders).			
Base	*ARObject*			
Subclasses	*AtpDefinition*, BswDistinguishedPartition, *BswModuleCallPoint*, BswModuleClientServerEntry, BswVariableAccess, CouplingPortTrafficClassAssignment, *DiagnosticEnvModeElement*, EthernetPriorityRegeneration, ExclusiveAreaNestingOrder, *HwDescriptionEntity*, *ImplementationProps*, LinSlaveConfigIdent, ModeTransition, *MultilanguageReferrable*, PncMappingIdent, *SingleLanguageReferrable*, SoConIPduIdentifier, TpConnectionIdent			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	Referrable (abstract)			
shortName	Identifier	1	attr	This specifies an identifying shortName for the object. It needs to be unique within its context and is intended for humans but even more for technical reference. **Stereotypes:** atpIdentityContributor **Tags:** xml.enforceMinMultiplicity=true xml.sequenceOffset=-100
shortName Fragment	ShortNameFragment	*	aggr	This specifies how the Referrable.shortName is composed of several shortNameFragments. **Tags:** atp.Splitkey=shortNameFragment xml.sequenceOffset=-90

Table C.51: Referrable

Class	Sdg			
Note	Sdg (SpecialDataGroup) is a generic model which can be used to keep arbitrary information which is not explicitly modeled in the meta-model. Sdg can have various contents as defined by sdgContentsType. Special Data should only be used moderately since all elements should be defined in the meta-model. Thereby SDG should be considered as a temporary solution when no explicit model is available. If an sdg Caption is available, it is possible to establish a reference to the sdg structure.			
Base	*ARObject*			
Aggregated by	AdminData.sdg, BuildActionEnvironment.sdg, BuildActionInvocator.sdg, BuildActionIoElement.sdg, File InfoComment.sdg, RptHook.sdg, SdgContents.sdg, VariationPoint.sdg			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
gid	NameToken	0..1	attr	This attributes specifies an identifier. Gid comes from the SGML/XML-Term "Generic Identifier" which is the element name in XML. The role of this attribute is the same as the name of an XML - element. **Tags:** xml.attribute=true
sdgCaption	SdgCaption	0..1	aggr	This aggregation allows to assign the properties of Identifiable to the sdg. By this, a shortName etc. can be assigned to the Sdg. **Stereotypes:** atpIdentityContributor **Tags:** xml.sequenceOffset=20
sdgContents Type	SdgContents	0..1	aggr	This is the content of the Sdg. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=30 xml.typeElement=false xml.typeWrapperElement=false

Table C.52: Sdg

Class	SenderReceiverInterface
Note	A sender/receiver interface declares a number of data elements to be sent and received. **Tags:** atp.recommendedPackage=PortInterfaces
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *DataInterface*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*
Aggregated by	ARPackage.element

▽

△

Class	**SenderReceiverInterface**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataElement	VariableDataPrototype	*	aggr	The data elements of this `SenderReceiverInterface`. **Tags:** atp.Splitkey=dataElement, dataElement.shortName
invalidationPolicy	InvalidationPolicy	*	aggr	InvalidationPolicy for a particular dataElement **Tags:** atp.Splitkey=invalidationPolicy
metaDataItemSet	MetaDataItemSet	*	aggr	This aggregation defines fixed sets of meta-data items associated with `dataElement`s of the enclosing `SenderReceiverInterface` **Tags:** atp.Splitkey=metaDataItemSet

Table C.53: SenderReceiverInterface

Class	***ServiceNeeds*** (abstract)			
Note	This expresses the abstract needs that a Software Component or Basic Software Module has on the configuration of an AUTOSAR Service to which it will be connected. "Abstract needs" means that the model abstracts from the Configuration Parameters of the underlying Basic Software.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	BswMgrNeeds, ChargeManagerNeeds, ComMgrUserNeeds, CryptoKeyManagementNeeds, CryptoServiceJobNeeds, CryptoServiceNeeds, *DiagnosticCapabilityElement*, DltUserNeeds, *DoIpServiceNeeds*, EcuStateMgrUserNeeds, ErrorTracerNeeds, FunctionInhibitionAvailabilityNeeds, FunctionInhibitionNeeds, GeneralPurposeTimerServiceNeeds, GlobalSupervisionNeeds, IdsMgrCustomTimestampNeeds, IdsMgrNeeds, IndicatorStatusNeeds, J1939DcmDm19Support, J1939RmIncomingRequestServiceNeeds, J1939RmOutgoingRequestServiceNeeds, NvBlockNeeds, SecureOnBoardCommunicationNeeds, SupervisedEntityCheckpointNeeds, SupervisedEntityNeeds, SyncTimeBaseMgrUserNeeds, VendorSpecificServiceNeeds			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table C.54: ServiceNeeds

Class	**ServiceSwComponentType**			
Note	ServiceSwComponentType is used for configuring services for a given ECU. Instances of this class are only to be created in ECU Configuration phase for the specific purpose of the service configuration. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, *ARObject*, *AtomicSwComponentType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table C.55: ServiceSwComponentType

Class	**SwBaseType**
Note	This meta-class represents a base type used within ECU software. **Tags:** atp.recommendedPackage=BaseTypes
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *BaseType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*

▽

△

Class	SwBaseType			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table C.56: SwBaseType

Class	SwComponentPrototype			
Note	Role of a software component within a composition.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, CompositionSwComponentType.component			
Attribute	Type	Mult.	Kind	Note
type	SwComponentType	0..1	tref	Type of the instance. **Stereotypes:** isOfType **Tags:** atp.Splitkey=type

Table C.57: SwComponentPrototype

Class	***SwComponentType*** (abstract)			
Note	Base class for AUTOSAR software components.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	*AtomicSwComponentType*, CompositionSwComponentType, ParameterSwComponentType			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
consistency Needs	ConsistencyNeeds	*	aggr	This represents the collection of `ConsistencyNeeds` owned by the enclosing `SwComponentType`. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=consistencyNeeds, consistencyNeeds.shortName, consistencyNeeds.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
port	PortPrototype	*	aggr	The `PortPrototype`s through which this `SwComponentType` can communicate. The aggregation of `PortPrototype` is subject to variability with the purpose to support the conditional existence of `PortPrototype`s. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=port, port.shortName, port.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
portGroup	PortGroup	*	aggr	A port group being part of this component. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=portGroup, portGroup.shortName, portGroup.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	***SwComponentType*** (abstract)			
swcMapping Constraint	SwComponentMapping Constraints	*	ref	Reference to constraints that are valid for this Sw ComponentType. **Tags:** atp.Splitkey=swcMappingConstraint This Attribute is only used by the AUTOSAR Classic Platform.
swComponent Documentation	SwComponent Documentation	0..1	aggr	This adds a documentation to the `SwComponentType`. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swComponentDocumentation, sw ComponentDocumentation.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=-10
unitGroup	UnitGroup	*	ref	This allows for the specification of which `UnitGroup`s are relevant in the context of referencing `SwComponentType`. **Tags:** atp.Splitkey=unitGroup This Attribute is only used by the AUTOSAR Classic Platform.

Table C.58: SwComponentType

Class	**SwSystemconst**			
Note	This element defines a system constant which serves an input to select a particular variation point. In particular a system constant serves as an operand of the binding function (swSyscond) in a Variation point. Note that the binding process can only happen if a value was assigned to to the referenced system constants. **Tags:** atp.recommendedPackage=SwSystemconsts			
Base	*ARElement*, *ARObject*, *AtpDefinition*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
swDataDef Props	SwDataDefProps	0..1	aggr	This denotes the data definition properties of the system constant. This supports to express the limits and optionally a conversion within the internal to physical values by a compu method. **Tags:** atp.Splitkey=swDataDefProps xml.sequenceOffset=40

Table C.59: SwSystemconst

Class	**System**
Note	The top level element of the System Description. The System description defines five major elements: Topology, Software, Communication, Mapping and Mapping Constraints. The System element directly aggregates the elements describing the Software, Mapping and Mapping Constraints; it contains a reference to an ASAM FIBEX description specifying Communication and Topology. **Tags:** atp.recommendedPackage=Systems
Base	*ARElement*, *ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*

▽

△

Class	System			
Aggregated by	ARPackage.element, *AtpClassifier*.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
clientId DefinitionSet	ClientIdDefinitionSet	*	ref	Set of Client Identifiers that are used for inter-ECU client-server communication in the System. **Tags:** atp.Splitkey=clientIdDefinitionSet This Attribute is only used by the AUTOSAR Classic Platform.
containerIPdu HeaderByte Order	ByteOrderEnum	0..1	attr	Defines the byteOrder of the header in ContainerIPdus. This Attribute is only used by the AUTOSAR Classic Platform.
ecuExtract Version	RevisionLabelString	0..1	attr	Version number of the Ecu Extract. This Attribute is only used by the AUTOSAR Classic Platform.
fibexElement	FibexElement	*	ref	Reference to ASAM FIBEX elements specifying Communication and Topology. All Fibex Elements used within a System Description shall be referenced from the System Element. atpVariation: In order to describe a product-line, all Fibex Elements can be optional. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=fibexElement, fibexElement.fibexElement, fibexElement.variationPoint.shortLabel vh.latestBindingTime=postBuild
interpolation Routine MappingSet	InterpolationRoutine MappingSet	*	ref	This reference identifies the InterpolationRoutineMapping Sets that are relevant in the context of the enclosing System. **Tags:** atp.Splitkey=interpolationRoutineMappingSet This Attribute is only used by the AUTOSAR Classic Platform.
j1939Shared AddressCluster	J1939SharedAddress Cluster	*	aggr	Collection of J1939Clusters that share a common address space for the routing of messages. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=j1939SharedAddressCluster, j1939Shared AddressCluster.shortName, j1939SharedAddress Cluster.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	System			
mapping	SystemMapping	*	aggr	Aggregation of all mapping aspects (mapping of SW components to ECUs, mapping of data elements to signals, and mapping constraints). In order to support OEM / Tier 1 interaction and shared development for one common System this aggregation is atpSplitable and atpVariation. The content of System Mapping can be provided by several parties using different names for the SystemMapping. This element is not required when the System description is used for a network-only use-case. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=mapping, mapping.shortName, mapping.variationPoint.shortLabel vh.latestBindingTime=postBuild
pncVector Length	PositiveInteger	0..1	attr	Length of the partial networking request release information vector (in bytes).
pncVectorOffset	PositiveInteger	0..1	attr	Absolute offset (with respect to the NM-PDU) of the partial networking request release information vector that is defined in bytes as an index starting with 0.
rootSoftware Composition	RootSwComposition Prototype	0..1	aggr	Aggregation of the root software composition, containing all software components in the System in a hierarchical structure. This element is not required when the System description is used for a network-only use-case. atpVariation: The RootSwCompositionPrototype can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=rootSoftwareComposition, rootSoftware Composition.shortName, rootSoftware Composition.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
swCluster	CpSoftwareCluster	*	ref	CP Software Clusters of this System **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swCluster, swCluster.cpSoftwareCluster, sw Cluster.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
systemCom SpecDefinition	SystemComSpec DefinitionSet	*	ref	Reference to the set of ComSpec definitions that are used for inter-ECU communication in the System. **Tags:** atp.Splitkey=systemComSpecDefinition
system Documentation	Chapter	*	aggr	Possibility to provide additional documentation while defining the System. The System documentation can be composed of several chapters. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=systemDocumentation, system Documentation.shortName, system Documentation.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=-10 This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	System			
systemVersion	RevisionLabelString	0..1	attr	Version number of the System Description.

Table C.60: System

Class	SystemSignal			
Note	The system signal represents the communication system's view of data exchanged between SW components which reside on different ECUs. The system signals allow to represent this communication in a flattened structure, with exactly one system signal defined for each data element prototype sent and received by connected SW component instances. **Tags:** atp.recommendedPackage=SystemSignals			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dynamicLength	Boolean	0..1	attr	The length of dynamic length signals is variable in run-time. Only a maximum length of such a signal is specified in the configuration (attribute length in ISignal element).
physicalProps	SwDataDefProps	0..1	aggr	Specification of the physical representation. **Tags:** atp.Splitkey=physicalProps

Table C.61: SystemSignal

Class	Unit			
Note	This is a physical measurement unit. All units that might be defined should stem from SI units. In order to convert one unit into another factor and offset are defined. For the calculation from SI-unit to the defined unit the factor (factorSiToUnit) and the offset (offsetSiTo Unit) are applied as follows: x [{unit}] := y * {siUnit} * factorSiToUnit [{unit}/{siUnit}] + offsetSiToUnit [{unit}] For the calculation from a unit to SI-unit the reciprocal of the factor (factorSiToUnit) and the negation of the offset (offsetSiToUnit) are applied. y [{siUnit}] := (x*{unit} - offsetSiToUnit [{unit}]) / factorSiToUnit [{unit}/{siUnit}] **Tags:** atp.recommendedPackage=Units			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
displayName	SingleLanguageUnit Names	0..1	aggr	This specifies how the unit shall be displayed in documents or in user interfaces of tools.The displayName corresponds to the Unit.Display in an ASAM MCD-2MC file. **Tags:** atp.Splitkey=displayName xml.sequenceOffset=20
factorSiToUnit	Float	0..1	attr	This is the factor for the conversion from SI Units to units. The inverse is used for conversion from units to SI Units. **Tags:** xml.sequenceOffset=30
offsetSiToUnit	Float	0..1	attr	This is the offset for the conversion from and to siUnits. **Tags:** xml.sequenceOffset=40

▽

△

Class	Unit			
physical Dimension	PhysicalDimension	0..1	ref	This association represents the physical dimension to which the unit belongs to. Note that only values with units of the same physical dimensions might be converted. **Tags:** atp.Splitkey=physicalDimension xml.sequenceOffset=50

Table C.62: Unit

Class	*ValueSpecification* (abstract)			
Note	Base class for expressions leading to a value which can be used to initialize a data object.			
Base	*ARObject*			
Subclasses	*AbstractRuleBasedValueSpecification*, ApplicationValueSpecification, *CompositeValueSpecification*, ConstantReference, NotAvailableValueSpecification, NumericalValueSpecification, ReferenceValue Specification, TextValueSpecification			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, ConstantSpecification.valueSpec, CryptoService Key.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnvDataElement Condition.compareValue, DiagnosticEnvSovdDataCondition.compareValue, FieldSenderComSpec.init Value, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, Nv ProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
shortLabel	Identifier	0..1	attr	This can be used to identify particular value specifications for human readers, for example elements of a record type.

Table C.63: ValueSpecification

Class	VariationPoint			
Note	This meta-class represents the ability to express a "structural variation point". The container of the variation point is part of the selected variant if swSyscond evaluates to true and each postBuildVariant Criterion is fulfilled.			
Base	*ARObject*			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
blueprint Condition	DocumentationBlock	0..1	aggr	This represents a description that documents how the variation point shall be resolved when deriving objects from the blueprint. Note that variationPoints are not allowed within a blueprintCondition. **Tags:** xml.sequenceOffset=28
desc	MultiLanguageOverview Paragraph	0..1	aggr	This allows to describe shortly the purpose of the variation point. **Tags:** xml.sequenceOffset=20

▽

△

Class	VariationPoint			
formalBlueprint Generator	BlueprintGenerator	0..1	aggr	This represents a description that documents how the variation point shall be resolved when deriving objects from the blueprint by using ARMQL. Note that variationPoints are not allowed within a formal BlueprintGenerator. **Tags:** atp.Status=draft xml.sequenceOffset=30
postBuildVariant Condition	PostBuildVariant Condition	*	aggr	This is the set of post build variant conditions which all shall be fulfilled in order to (postbuild) bind the variation point. **Tags:** xml.sequenceOffset=40
sdg	Sdg	0..1	aggr	An optional special data group is attached to every variation point. These data can be used by external software systems to attach application specific data. For example, a variant management system might add an identifier, an URL or a specific classifier. **Tags:** xml.sequenceOffset=50
shortLabel	Identifier	0..1	attr	This provides a name to the particular variation point to support the RTE generator. It is necessary for supporting splitable aggregations and if binding time is later than codeGenerationTime, as well as some RTE conditions. It needs to be unique with in the enclosing Identifiables with the same ShortName. **Stereotypes:** atpIdentityContributor **Tags:** xml.sequenceOffset=10
swSyscond	ConditionByFormula	0..1	aggr	This condition acts as Binding Function for the Variation Point. Note that the multiplicity is 0..1 in order to support pure postBuild variants. **Tags:** xml.sequenceOffset=30

Table C.64: VariationPoint

D Glossary

The content of this appendix chapter is *informative* in nature and shall **not** be considered as *normative* content.

Artifact This is a Work Product Definition that provides a description and definition for tangible work product types. Artifacts may be composed of other artifacts ([24]).

At a high level, an artifact is represented as a single conceptual file.

AUTOSAR Tool This is a software tool which supports one or more tasks defined as AUTOSAR tasks in the methodology. Depending on the supported tasks, an AUTOSAR tool can act as an authoring tool, a converter tool, a processor tool or as a combination of those (see separate definitions).

AUTOSAR Authoring Tool An AUTOSAR Tool used to create and modify AUTOSAR XML Descriptions. Example: System Description Editor.

AUTOSAR Converter Tool An AUTOSAR Tool used to create AUTOSAR XML files by converting information from other AUTOSAR XML files. Example: ECU Flattener

AUTOSAR Definition This is the definition of parameters which can have values. One could say that the parameter values are Instances of the definitions. But in the meta model hierarchy of AUTOSAR, definitions are also instances of the meta model and therefore considered as a description. Examples for AUTOSAR definitions are: `PortPrototype`, `PostBuildVariantCriterion`, `SwSystemconst`.

AUTOSAR XML Description In AUTOSAR this means "filled Template". In fact an AUTOSAR XML description is the XML representation of an AUTOSAR model.

The AUTOSAR XML description can consist of several files. Each individual file represents an AUTOSAR partial model and shall validate successfully against the AUTOSAR XML schema.

AUTOSAR Meta-Model This is an UML2.0 model that defines the language for describing AUTOSAR systems. The AUTOSAR meta-model is an UML representation of the AUTOSAR templates. UML2.0 class diagrams are used to describe the attributes and their interrelationships. Stereotypes, UML tags and OCL expressions (object constraint language) are used for defining specific semantics and constraints.

AUTOSAR Meta-Model Tool The AUTOSAR Meta-Model Tool is the tool that generates different views (class tables, list of constraints, diagrams, XML Schema etc.) on the AUTOSAR meta-model.

AUTOSAR Model This is a representation of an AUTOSAR product. The AUTOSAR model represents aspects suitable to the intended use according to the AUTOSAR methodology.

Strictly speaking, this is an instance of the AUTOSAR meta-model. The information contained in the AUTOSAR model can be anything that is representable according to the AUTOSAR meta-model.

AUTOSAR Partial Model In AUTOSAR, the possible partitioning of models is marked in the meta-model by ≪`atpSplitable`≫. One partial model is represented in an AUTOSAR XML description by one file. The partial model does not need to fulfill all semantic constraints applicable to an AUTOSAR model.

AUTOSAR Processor Tool An AUTOSAR Tool used to create non-AUTOSAR files by processing information from AUTOSAR XML files. Example: RTE Generator

AUTOSAR Specification Element An AUTOSAR Specification Element is a named element that is part of an AUTOSAR specification. Examples: requirement, constraint, specification item, class or attribute in the meta model, methodology, deliverable, methodology activity, model element, bsw module etc.

AUTOSAR Template The term "Template" is used in AUTOSAR to describe the format different kinds of descriptions. The term template comes from the idea, that

AUTOSAR defines a kind of form which shall be filled out in order to describe a model. The filled form is then called the description.

In fact the AUTOSAR templates are now defined as a meta-model.

AUTOSAR Validation Tool A specialized `AUTOSAR Tool` which is able to check an AUTOSAR model against the rules defined by a profile.

AUTOSAR XML Schema This is a W3C XML schema that defines the language for exchanging AUTOSAR models. This Schema is derived from the AUTOSAR meta-model. The AUTOSAR XML Schema defines the AUTOSAR data exchange format.

Blueprint This is a model from which other models can be derived by copy and refinement. Note that in contrast to meta model resp. types, this process is *not* an instantiation.

Instance Generally this is a particular exemplar of a model or of a type.

Life Cycle Life Cycle is the course of development/evolutionary stages of a model element during its life time.

Meta-Model This defines the building blocks of a model. In that sense, a Meta-Model represents the language for building models.

Meta-Data This includes pertinent information about data, including information about the authorship, versioning, access-rights, timestamps etc.

Model A Model is an simplified representation of reality. The model represents the aspects suitable for an intended purpose.

Partial Model This is a part of a model which is intended to be persisted in one particular artifact.

Pattern in GST This is an approach to simplify the definition of the meta model by applying a model transformation. This transformation creates an enhanced model out of an annotated model.

Property A property is a structural feature of an object. As an example a “connector” has the properties “receive port” and “send port”.

Properties are made variant by the ≪`atpVariation`≫.

Prototype This is the implementation of a role of a type within the definition of another type. In other words a type may contain Prototypes that in turn are typed by “Types”. Each one of these prototypes becomes an instance when this type is instantiated.

Type A type provides features that can appear in various roles of this type.

Value This is a particular value assigned to a “Definition”.

Variability Variability of a system is its quality to describe a set of variants. These variants are characterized by variant specific property settings and / or selections.

As an example, such a system property selection manifests itself in a particular “receive port” for a connection.

This is implemented using the ≪atpVariation≫.

Variant A system variant is a concrete realization of a system, so that all its properties have been set respectively selected. The software system has no variability anymore with respect to the binding time.

This is implemented using EvaluatedVariantSet.

Variation Binding A variant is the result of a variation binding process that resolves the variability of the system by assigning particular values/selections to all the system’s properties.

This is implemented by VariationPoint.

Variation Binding Time The variation binding time determines the step in the methodology at which the variability given by a set of variable properties is resolved.

This is implemented by vh.latestBindingTime at the related properties.

Variation Definition Time The variation definition time determines the step in the methodology at which the variation points are defined.

Variation Point A variation point indicates that a property is subject to variation. Furthermore, it is associated with a condition and a binding time which define the system context for the selection / setting of a concrete variant.

This is implemented by VariationPoint.

E Change history of AUTOSAR traceable items

The content of this appendix chapter is *informative* in nature and shall **not** be considered as *normative* content.

E.1 Traceable item history of this document according to AUTOSAR Release R4.2.1

E.1.1 Added Specification Items in R4.2.1

Number	Heading
[TPS_DEXT_01000]	AUTOSAR diagnostics supports two kinds of data identifiers
[TPS_DEXT_01001]	Definition of a fixed-sized array
[TPS_DEXT_01002]	Definition of a variable-sized array
[TPS_DEXT_01003]	`DiagnosticContributionSet` is the central part of the `DiagnosticExtract`
[TPS_DEXT_01004]	`DiagnosticContributionSet` defines the scope of the `DiagnosticExtract`
[TPS_DEXT_01005]	`DiagnosticContributionSet` can exist independently
[TPS_DEXT_01006]	The role of `DiagnosticServiceTable`s in the context of a `DiagnosticContributionSet`
[TPS_DEXT_01007]	Common properties of a `DiagnosticExtract`
[TPS_DEXT_01008]	`DiagnosticContributionSet` defines the scope for the application of the common diagnostic properties
[TPS_DEXT_01009]	Limited support for the configuration of custom diagnostic services
[TPS_DEXT_01010]	Configuration of custom diagnostic services
[TPS_DEXT_01011]	Semantics of `DiagnosticSession.id`
[TPS_DEXT_01012]	Rationale for the modeling of the multiplicity of `DiagnosticAccessPermission.securityLevel`
[TPS_DEXT_01013]	Specification of sub-functions by means of attribute `DiagnosticServiceInstance.category`
[TPS_DEXT_01014]	Possible values of the `category` attribute for diagnostic services
[TPS_DEXT_01015]	Meaning of attributes of `DiagnosticIOControl`
[TPS_DEXT_01016]	The capability returnControlToEcu
[TPS_DEXT_01017]	Meaning of `DiagnosticIOControl.dataIdentifier`
[TPS_DEXT_01018]	`InputOutput Control` does not define any sub-functions
[TPS_DEXT_01019]	Correspondence of `category` values to numerical values mentioned in the ISO 14229-1
[TPS_DEXT_01020]	Manufacturer-specific values for sub-functions of service `EcuReset`
[TPS_DEXT_01021]	Semantics of `DiagnosticEcuReset.customSubFunctionNumber`
[TPS_DEXT_01022]	`ClearDiagnosticInformation` does not define any sub-functions

▽

△

Number	Heading
[TPS_DEXT_01023]	`WriteMemoryByAddress` does not define any sub-functions
[TPS_DEXT_01024]	`ReadMemoryByAddress` does not define any sub-functions
[TPS_DEXT_01025]	`TransferExit` does not define any sub-functions
[TPS_DEXT_01026]	`DataTransfer` does not define any sub-functions
[TPS_DEXT_01027]	`RequestDownload` does not define any sub-functions
[TPS_DEXT_01028]	`RequestUpload` does not define any sub-functions
[TPS_DEXT_01029]	Correspondence of `category` values to numerical values mentioned in the ISO 14229-1
[TPS_DEXT_01030]	Manufacturer-specific values for sub-functions of service `CommunicationControl`
[TPS_DEXT_01031]	Semantics of `DiagnosticComControl.customSubFunctionNumber`
[TPS_DEXT_01032]	Impact of the `DiagnosticComControlClass` on the state management for `CommunicationCluster`s
[TPS_DEXT_01033]	Semantics of triggers in the context of a `DiagnosticResponseOnEvent`
[TPS_DEXT_01034]	Sub-functions of the service `ReadDTCInformation`
[TPS_DEXT_01035]	Existence of `DiagnosticRoutine.stop` and `DiagnosticRoutine.requestResult`
[TPS_DEXT_01036]	Work-flow within the execution of the diagnostic service `SecurityAccess`
[TPS_DEXT_01037]	Semantics of `DiagnosticSecurityAccess.requestSeedId`
[TPS_DEXT_01038]	Motivation for making the reference `DiagnosticSecurityAccess.securityLevel`≪`atpSplitable`≫
[TPS_DEXT_01039]	Identification of the sub-function of `DiagnosticSessionControl`
[TPS_DEXT_01040]	Use case where the `DiagnosticExtract` refers to software-components
[TPS_DEXT_01041]	Semantics of attribute `DiagnosticServiceDataMapping.diagnosticDataElement`
[TPS_DEXT_01042]	Dem uses `DiagnosticServiceDataMapping`
[TPS_DEXT_01043]	Purpose of `DiagnosticServiceSwMapping`
[TPS_DEXT_01044]	`BswServiceDependency` needs to act as the target of a reference
[TPS_DEXT_01045]	Supported diagnostic services
[TPS_DEXT_01046]	ECU configuration is not suitable to be exchanged between partners in an ECU development project
[TPS_DEXT_01047]	Differences in the development processes for diagnostics at automotive OEMs and ECU suppliers
[TPS_DEXT_01048]	Actual algorithm for the diagnostic event debouncing
[TPS_DEXT_01049]	Consistency of `DiagnosticServiceSwMapping` with respect to routine IDs
[TPS_DEXT_01050]	Consistency of `DiagnosticServiceSwMapping` with respect to data IDs
[TPS_DEXT_01051]	Consistency of `DiagnosticServiceSwMapping` with respect to data IDs
[TPS_DEXT_01052]	Existence of attribute `DiagnosticServiceInstance.accessPermission`
[TPS_DEXT_01053]	Existence of `DiagnosticSecurityAccess.securityLevel`
[TPS_DEXT_01054]	Existence of `DiagnosticDataByIdentifier.dataIdentifier`

▽

△

Number	Heading
[TPS_DEXT_01055]	Standardized values of DiagnosticContributionSet.category
[TPS_DEXT_01056]	Applicable values for DiagnosticEcuReset.category
[TPS_DEXT_01057]	Allowed values of DiagnosticComControl.category
[TPS_DEXT_01058]	Standardized values for DiagnosticDynamicallyDefineDataIdentifier.category
[TPS_DEXT_01059]	Applicable values for DiagnosticPeriodicRate.category
[TPS_DEXT_01060]	Applicable values for DiagnosticReadDTCInformation.category
[TPS_DEXT_01061]	Supported scenarios for the definition of *access permission*
[TPS_DEXT_01062]	Existence of DiagnosticServiceClass.accessPermissionValidity in an incomplete model
[TPS_DEXT_01063]	Existence of DiagnosticServiceClass.accessPermissionValidity in a complete model
[TPS_DEXT_01064]	Textually formulated content attached to DiagnosticTroubleCode
[TPS_DEXT_01065]	Different approaches to provide semi-formal textual content attached to a DiagnosticTroubleCode
[TPS_DEXT_01066]	Standardized values of DiagnosticTroubleCode.introduction.trace
[TPS_DEXT_01067]	Textually formulated content attached to DiagnosticEvent
[TPS_DEXT_01068]	Textual description with respect to the DiagnosticEvent
[TPS_DEXT_01069]	Standardized values of DiagnosticEvent.introduction.structuredReq
[TPS_DEXT_01070]	Description of textually semi-formal formulated pre- and post-conditions for the validity of DiagnosticAccessPermission
[TPS_DEXT_01071]	Standardized values of DiagnosticAccessPermission.introduction.structuredReq
[TPS_DEXT_01072]	Purpose of attribute DiagnosticDataIdentifier.representsVin
[TPS_DEXT_01073]	Diagnostic properties that are specific to an individual EcuInstance
[TPS_DEXT_01074]	Difference between the attributes DiagnosticComControl.specificChannel and DiagnosticComControl.subNodeChannel
[TPS_DEXT_01075]	standardized values for the attribute DiagnosticControlDTCSetting.category
[TPS_DEXT_01076]	Identification of sub-functions of diagnostic service ControlDTCSetting
[TPS_DEXT_01077]	Modeling of DiagnosticRoutine
[TPS_DEXT_01078]	Not possible to use the attribute category for the identification of the sub-function of diagnostic service RoutineControl
[TPS_DEXT_01079]	Modeling of the arguments to a DiagnosticRoutine
[TPS_DEXT_01080]	Diagnostic Routine needs to be started
[TPS_DEXT_01081]	Modeling of DiagnosticSessionControl
[TPS_DEXT_01082]	Existence of DiagnosticSessionControl.diagnosticSession
[TPS_DEXT_01083]	Semantics of a DiagnosticEvent
[TPS_DEXT_01084]	DiagnosticEvent can be connected to one or multiple indicators

▽

△

Number	Heading
[TPS_DEXT_01085]	Semantics of DiagnosticConditionGroups
[TPS_DEXT_01086]	Reference to DiagnosticOperationCycle
[TPS_DEXT_01087]	Semantics of DiagnosticOperationCycle
[TPS_DEXT_01088]	Semantics of DiagnosticRoutine.id
[TPS_DEXT_01089]	Definition of an *identifier* of a DiagnosticIOControl
[TPS_DEXT_01090]	Diagnostic service RequestFileTransfer does not define any sub-functions
[TPS_DEXT_03000]	ISO 14229-1 reserves values of DiagnosticTroubleCodeGroup.groupNumber
[TPS_DEXT_03001]	Different types of conditions
[TPS_DEXT_03002]	Two kind of mappings
[TPS_DEXT_03003]	Semantics of DiagnosticEventToTroubleCodeUdsMapping
[TPS_DEXT_03004]	DiagnosticEvent and DiagnosticDebounceAlgorithmProps
[TPS_DEXT_03005]	Existence of DiagnosticEventToDebounceAlgorithmMapping
[TPS_DEXT_03006]	Values of the individual DiagnosticStorageConditions
[TPS_DEXT_03007]	Semantics of DiagnosticEventPortMapping
[TPS_DEXT_03008]	Semantics of DiagnosticExtendedDataRecord
[TPS_DEXT_03009]	Semantics of DiagnosticFreezeFrame
[TPS_DEXT_03010]	Combination of DiagnosticConditions to DiagnosticConditionGroups
[TPS_DEXT_03011]	Clearing request for a DiagnosticEvent
[TPS_DEXT_03012]	Three kinds of DTCs
[TPS_DEXT_03013]	Common properties of a DTC
[TPS_DEXT_03014]	Semantics of DiagnosticTroubleCodeGroup
[TPS_DEXT_03015]	EnableConditions have to be put into a DiagnosticEnableConditionGroup
[TPS_DEXT_03016]	StorageConditions have to be put into a DiagnosticStorageConditionGroup
[TPS_DEXT_03017]	Semantics of DiagnosticOperationCyclePortMapping
[TPS_DEXT_03018]	Semantics of DiagnosticEnableConditionPortMapping
[TPS_DEXT_03019]	Semantics of DiagnosticStorageConditionPortMapping
[TPS_DEXT_03020]	Semantics of DiagnosticDemProvidedDataMapping
[TPS_DEXT_03021]	Aging
[TPS_DEXT_03022]	Different kinds of DiagnosticIndicators

Table E.1: Added Specification Items in 4.2.1

E.1.2 Added Constraints in R4.2.1

Number	Heading
[constr_1324]	Existence of attribute DiagnosticDataIdentifier.representsVin
[constr_1325]	Allowed attributes of SwDataDefProps for DiagnosticDataElement.swDataDefProps
[constr_1326]	Existence of a variable-sized array
[constr_1327]	Multiplicity of DiagnosticContributionSet.ecuInstance
[constr_1328]	Consistency of DiagnosticContributionSet.ecuInstance and DiagnosticServiceTable.ecuInstance
[constr_1329]	Existence of concrete sub-classes of DiagnosticServiceClass in the context created by a DiagnosticContributionSet
[constr_1330]	Custom service identifier shall not overlap with standardized service identifiers
[constr_1331]	Existence of DiagnosticEcuReset.customSubFunctionNumber
[constr_1332]	Value range for DiagnosticEcuReset.customSubFunctionNumber
[constr_1333]	Existence of DiagnosticMemoryIdentifier.memoryLowAddress and DiagnosticMemoryIdentifier.memoryHighAddress
[constr_1334]	Existence of DiagnosticComControl.customSubFunctionNumber
[constr_1335]	Possible values for DiagnosticComControl.customSubFunctionNumber
[constr_1336]	Applicable value range for DiagnosticComControlSpecificChannel.subnetNumber
[constr_1337]	Allowed value range for attribute DiagnosticComControlSubNodeChannel.subNodeNumber
[constr_1338]	Maximum number of aggregated DiagnosticReadDataByPeriodicIDClass.periodicRate
[constr_1339]	Existence of DiagnosticRoutine.start
[constr_1340]	Consistency of DiagnosticServiceSwMapping with respect to synchronously called DiagnosticRoutines
[constr_1341]	Consistency of DiagnosticServiceSwMapping with respect to asynchronously called DiagnosticRoutines
[constr_1342]	Possible values for DiagnosticSecurityAccess.requestSeedId
[constr_1343]	Simultaneous existence of the attributes DiagnosticServiceDataMapping.diagnosticDataElement and DiagnosticDataByIdentifier.dataIdentifier
[constr_1344]	Condition for the identification of data types of attributes DiagnosticServiceDataMapping.mappedDataElement and DiagnosticServiceDataMapping.diagnosticDataElement
[constr_1345]	DiagnosticDataElement shall not (finally) be aggregated by a DiagnosticRoutine
[constr_1346]	Allowed values of DiagnosticServiceSwMapping.serviceInstance
[constr_1347]	Existence of attributes of DiagnosticServiceSwMapping
[constr_1349]	Value of udsDtcValue shall be unique
[constr_1350]	Value of DiagnosticTroubleCodeGroup.groupNumber shall be unique

▽

△

Number	Heading
[constr_1351]	Value of DiagnosticTroubleCodeGroup.groupNumber
[constr_1352]	Existence of maxNumberFreezeFrameRecords vs. freezeFrame
[constr_1353]	Applicability of [constr_1352]
[constr_1354]	Existence of attribute DiagnosticTroubleCodeProps.freezeFrameContent
[constr_1355]	Value of recordNumber
[constr_1356]	Value of recordNumber shall be unique
[constr_1357]	Value of recordNumber
[constr_1358]	Value of recordNumber shall be unique
[constr_1359]	Existence of attribute DiagnosticDebounceAlgorithmProps.debounceCounterStorage
[constr_1360]	Usage of DiagEventDebounceMonitorInternal is not supported in the context of DiagnosticDebounceAlgorithmProps
[constr_1361]	Number of DiagnosticEventToEnableConditionGroupMapping elements per DiagnosticEvent
[constr_1362]	Number of DiagnosticEventToStorageConditionGroupMapping elements per DiagnosticEvent
[constr_1365]	Multiplicity of DiagnosticResponseOnEvent.event
[constr_1366]	Event ID in the context of diagnostic service ResponseOnEvent shall be unique
[constr_1376]	Multiplicity of reference DiagnosticTroubleCodeProps.memoryDestination
[constr_1377]	Existence of reference DiagnosticTroubleCodeProps.memoryDestination
[constr_1378]	Value of DiagnosticMemoryDestinationUserDefined.memoryId
[constr_1379]	Existence of DiagnosticMemoryDestinationPrimary
[constr_1380]	Existence of DiagnosticMemoryDestinationMirror

Table E.2: Added Constraints in R4.2.1

E.2 Traceable item history of this document according to AUTOSAR Release R4.2.2

E.2.1 Added Traceables in R4.2.2

none

E.2.2 Changed Traceables in R4.2.2

none

E.2.3 Deleted Traceables in R4.2.2

none

E.2.4 Added Constraints in R4.2.2

Number	Heading
[constr_1394]	Value of DiagnosticDataElement.maxNumberOfElements depending on its existence

Table E.3: Added Constraints in R4.2.2

E.2.5 Changed Constraints in R4.2.2

none

E.2.6 Deleted Constraints in R4.2.2

none

E.3 Traceable item history of this document according to AUTOSAR Release R4.3.0

E.3.1 Added Traceables in R4.3.0

Number	Heading
[TPS_DEXT_01091]	Relation between a DiagnosticServiceTable and one or more DiagnosticConnections
[TPS_DEXT_01092]	Semantics of DiagnosticParameterIdentifier
[TPS_DEXT_01093]	Definition of legislative freeze frame for WWWH-OBD
[TPS_DEXT_01094]	Semantics of meta-class DiagnosticTroubleCodeUdsToTroubleCodeObdMapping
[TPS_DEXT_01095]	Definition of “alias” diagnostic event for the creation of a Fim configuration in the diagnostic extract
[TPS_DEXT_01096]	Semantics of DiagnosticFunctionInhibitSource
[TPS_DEXT_01097]	Standardized value of StructuredReq.category for the modeling of DiagnosticFunctionInhibitSource
[TPS_DEXT_01098]	Semantics of attribute DiagnosticFunctionInhibitSource.event
[TPS_DEXT_01099]	Semantics of attribute DiagnosticFunctionInhibitSource.eventGroup
[TPS_DEXT_01100]	Consequence of the existence of DiagnosticFimAliasEventMapping
[TPS_DEXT_01101]	Consequence of the existence of DiagnosticFimAliasEventGroupMapping
[TPS_DEXT_01102]	Semantics of DiagnosticFimFunctionMapping

▽

△

Number	Heading
[TPS_DEXT_01103]	Semantics of meta-class DiagnosticJ1939SpnMapping
[TPS_DEXT_01104]	Difference between DiagnosticJ1939FreezeFrame and DiagnosticJ1939ExpandedFreezeFrame
[TPS_DEXT_01105]	Relation of DiagnosticJ1939Spn to DiagnosticJ1939FreezeFrame and DiagnosticJ1939ExpandedFreezeFrame
[TPS_DEXT_01106]	Relation of Controller Application to SPN
[TPS_DEXT_01107]	Definition of service-only DTC
[TPS_DEXT_01108]	Purpose of the DiagnosticJ1939SwMapping
[TPS_DEXT_01110]	Standardized values of DiagnosticIumprGroup.category
[TPS_DEXT_01111]	Legislative freeze frame for the OBD-II case
[TPS_DEXT_01112]	Definition of a diagnostic trouble code for the implementation of WWH-OBD
[TPS_DEXT_01113]	Evaluation of a DiagnosticEnvConditionFormula
[TPS_DEXT_01114]	DiagnosticEnvConditionFormula that has no parts
[TPS_DEXT_01115]	DiagnosticEnvConditionFormula that has one part
[TPS_DEXT_01116]	DiagnosticEnvConditionFormula that has more than one part
[TPS_DEXT_01117]	Semantics of DiagnosticEnvConditionFormula.nrcValue
[TPS_DEXT_01118]	Semantics of DiagnosticEnvDataCondition
[TPS_DEXT_01119]	Semantics of DiagnosticEnvModeCondition
[TPS_DEXT_01120]	Comparison of the value of a ModeDeclarationGroupPrototype with a ModeDeclaration
[TPS_DEXT_01121]	Semantics of DiagnosticFunctionIdentifier
[TPS_DEXT_01122]	Indication whether a EcuInstance supports OBD
[TPS_DEXT_01124]	Semantics of meta-class DiagnosticProtocol
[TPS_DEXT_01125]	Support for diagnostic service RequestCurrentPowertrainDiagnosticData
[TPS_DEXT_01126]	Support of OBD service RequestPowertrainFreezeFrameData
[TPS_DEXT_01127]	Semantics of meta-class DiagnosticRequestEmissionRelatedDTC
[TPS_DEXT_01128]	Semantics of meta-class DiagnosticClearResetEmissionRelatedInfo
[TPS_DEXT_01129]	Support for OBD diagnostic service RequestOnBoardMonitoringTestResults
[TPS_DEXT_01130]	Support of OBD diagnostic service RequestControlOfOnBoardDevice
[TPS_DEXT_01131]	Support for OBD diagnostic service RequestVehicleInformation
[TPS_DEXT_01132]	Support for OBD diagnostic service RequestEmissionRelatedDiagnosticTroubleCodesPermanentStatus
[TPS_DEXT_01133]	Support for WWH-OBD within the diagnostic extract
[TPS_DEXT_01134]	Definition of a DiagnosticDataElement used in the context of a DID obtained by sender-receiver communication
[TPS_DEXT_01135]	Definition of a DiagnosticDataElement used in the context of a DID obtained by client/server communication
[TPS_DEXT_01136]	Definition of a DiagnosticDataElement used in the context of a diagnostic routine

▽

△

Number	Heading
[TPS_DEXT_01137]	Applicability of DiagnosticDataIdentifier.didSize
[TPS_DEXT_01138]	Applicability of DiagnosticDataIdentifier.supportInfoByte
[TPS_DEXT_01139]	Semantics of the references from DiagnosticAccessPermission

Table E.4: Added Traceabless in 4.3.0

E.3.2 Changed Traceables in R4.3.0

Number	Heading
[TPS_DEXT_01006]	The role of DiagnosticServiceTables
[TPS_DEXT_01052]	Existence of attribute DiagnosticServiceInstance.accessPermission
[TPS_DEXT_01060]	Applicable values for DiagnosticReadDTCInformation.category
[TPS_DEXT_01074]	Difference between the attributes DiagnosticComControlClass.specificChannel and DiagnosticComControlClass.subNodeChannel
[TPS_DEXT_03003]	Semantics of DiagnosticEventToTroubleCodeUdsMapping

Table E.5: Changed Traceables in R4.3.0

E.3.3 Deleted Traceables in R4.3.0

Number	Heading
[TPS_DEXT_01058]	Standardized values for DiagnosticDynamicallyDefineDataIdentifier.category
[TPS_DEXT_01059]	Applicable values for DiagnosticPeriodicRate.category

Table E.6: Deleted Traceables in R4.3.0

E.3.4 Added Constraints in R4.3.0

Number	Heading
[constr_1405]	Value of DiagnosticProtocol.serviceTable vs. DiagnosticServiceTable.protocolKind
[constr_1406]	DiagnosticServiceTable.diagnosticConnection vs. DiagnosticProtocol.diagnosticConnection
[constr_1411]	Existence of DiagnosticMemoryIdentifier.memoryHighAddressLabel vs. DiagnosticMemoryIdentifier.memoryHighAddress

▽

△

Number	Heading
[constr_1412]	Existence of DiagnosticMemoryIdentifier.memoryLowAddressLabel vs. DiagnosticMemoryIdentifier.memoryLowAddress
[constr_1419]	Value of DiagnosticSecurityLevel.accessDataRecordSize
[constr_1421]	Consistency of DiagnosticDynamicallyDefineDataIdentifierClass.sub-function
[constr_1435]	Debouncing in the presence of a DiagnosticEventPortMapping
[constr_1447]	Restrictions for the value of DiagnosticParameterIdentifier.id
[constr_1448]	Interval of DiagnosticParameterIdentifier.id
[constr_1449]	PID shall only carry a fixed-length collection of data
[constr_1450]	Service mapping for ODB mode 0x01 for DiagnosticParameterIdentifier
[constr_1451]	Service mapping for OBD mode 0x09 for DiagnosticInfoType
[constr_1452]	Service mapping for OBD mode 0x08 for DiagnosticInfoType
[constr_1453]	References from DiagnosticFunctionInhibitSource
[constr_1454]	DiagnosticFimFunctionMapping shall only reference a SwcService-Dependency that aggregates FunctionInhibitionNeeds
[constr_1455]	Relation of DiagnosticJ1939Node to J1939NmNode
[constr_1456]	Valid interval for attribute DiagnosticTroubleCodeJ1939.fmi
[constr_1457]	Service-only DTCs shall refer to a common memory section
[constr_1458]	Reference to DiagnosticMemoryDestination
[constr_1459]	Existence of attributes of DiagnosticTroubleCodeProps
[constr_1460]	Restrictions for the value of DiagnosticInfoType.id
[constr_1461]	Restrictions for the value of DiagnosticTestRoutineIdentifier.id
[constr_1462]	Restrictions for the value of DiagnosticTestResult.testIdentifier.id
[constr_1464]	Allowed value range of DiagnosticEnvConditionFormula.nrcValue
[constr_1465]	Allowed values of compareType in the context of a DiagnosticEnvData-Condition
[constr_1466]	Allowed values of compareType in the context of a DiagnosticEnvMode-Condition
[constr_1467]	References in DiagnosticEnvModeCondition
[constr_1470]	Value of DiagnosticParameter.bitOffset
[constr_1471]	Existence of DiagnosticDataIdentifier.didSize
[constr_1472]	Existence of DiagnosticDataIdentifier.supportInfoByte

Table E.7: Added Constraints in R4.3.0

E.3.5 Changed Constraints in R4.3.0

Number	Heading
[constr_1325]	Allowed attributes of SwDataDefProps for DiagnosticDataElement.swDataDefProps
[constr_1327]	Multiplicity of DiagnosticEcuInstanceProps.ecuInstance
[constr_1328]	Consistency of DiagnosticEcuInstanceProps.ecuInstance and DiagnosticServiceTable.ecuInstance

Table E.8: Changed Constraints in R4.3.0

E.3.6 Deleted Constraints in R4.3.0

none

E.4 Traceable item history of this document according to AUTOSAR Release R4.3.1

E.4.1 Added Specification Items in 4.3.1

Number	Heading
[TPS_DEXT_01140]	Values contained in DiagnosticExtract shall be taken for the derivation of basic software modules

Table E.9: Added Specification Items in 4.3.1

E.4.2 Changed Specification Items in 4.3.1

Number	Heading
[TPS_DEXT_01087]	Semantics of DiagnosticOperationCycle

Table E.10: Changed Specification Items in 4.3.1

E.4.3 Deleted Specification Items in 4.3.1

Number	Heading
[TPS_DEXT_01073]	Diagnostic properties that are specific to an individual EcuInstance

Table E.11: Deleted Specification Items in 4.3.1

E.4.4 Added Constraints in 4.3.1

Number	Heading
[constr_1509]	`extendedDataRecord.recordNumber` shall be unique within primary fault memory
[constr_1510]	`extendedDataRecord.recordNumber` shall be unique within mirror fault memory
[constr_1511]	`extendedDataRecord.recordNumber` shall be unique within user-defined fault memory
[constr_1512]	`freezeFrame.recordNumber` shall be unique within primary fault memory
[constr_1513]	`freezeFrame.recordNumber` shall be unique within mirror fault memory
[constr_1514]	`freezeFrame.recordNumber` shall be unique within user-defined fault memory
[constr_1515]	Reference from `DiagnosticRoutineControl` to `DiagnosticAccessPermission` has no meaning

Table E.12: Added Constraints in 4.3.1

E.4.5 Changed Constraints in 4.3.1

Number	Heading
[constr_1325]	Allowed attributes of `SwDataDefProps` for `DiagnosticDataElement.swDataDefProps`
[constr_1450]	Service mapping for ODB mode 0x01 for `DiagnosticParameterIdentifier`
[constr_1451]	Service mapping for OBD mode 0x09 for `DiagnosticInfoType`

Table E.13: Changed Constraints in 4.3.1

E.4.6 Deleted Constraints in 4.3.1

Number	Heading
[constr_1356]	Value of `recordNumber` shall be unique
[constr_1358]	Value of `recordNumber` shall be unique

Table E.14: Deleted Constraints in 4.3.1

E.5 Traceable item history of this document according to AUTOSAR Release R4.4.0

E.5.1 Added Specification Items in 4.4.0

Number	Heading
[TPS_DEXT_01141]	Definition of a collection of test results
[TPS_DEXT_01142]	Standardized values for DiagnosticDemProvidedDataMapping.dataProvider
[TPS_DEXT_01143]	Definition of a custom trigger for an extended data record
[TPS_DEXT_01144]	Definition of a custom trigger for a freeze frame
[TPS_DEXT_01145]	Semantics of meta-class DiagnosticEventToTroubleCodeJ1939Mapping
[TPS_DEXT_01146]	Support for service ReadScalingDataByIdentifier (0x24)

Table E.15: Added Specification Items in 4.4.0

E.5.2 Changed Specification Items in 4.4.0

Number	Heading
[TPS_DEXT_01004]	DiagnosticContributionSet defines the scope of the DiagnosticExtract
[TPS_DEXT_01041]	Semantics of attribute DiagnosticServiceDataMapping.diagnosticDataElement
[TPS_DEXT_01055]	Standardized values of DiagnosticContributionSet.category
[TPS_DEXT_01079]	Modeling of the arguments to a DiagnosticRoutine
[TPS_DEXT_03011]	Clearing request for a DiagnosticEvent
[TPS_DEXT_03020]	Semantics of DiagnosticDemProvidedDataMapping

Table E.16: Changed Specification Items in 4.4.0

E.5.3 Deleted Specification Items in 4.4.0

Number	Heading
[TPS_DEXT_01137]	Applicability of DiagnosticDataIdentifier.didSize

Table E.17: Deleted Specification Items in 4.4.0

E.5.4 Added Constraints in 4.4.0

Number	Heading
[constr_1552]	DiagnosticDataIdentifier referenced by DiagnosticDataIdentifierSet
[constr_1584]	DiagnosticDataElement shall not be used more than once in I/O Control instance
[constr_1590]	DiagnosticEvent referenced in the role masterEvent or slaveEvent
[constr_1591]	DiagnosticEvent referenced as slaveEvent shall not be reported by diagnostic monitor
[constr_1612]	Reference from DiagnosticRoutineControl to DiagnosticAccessPermission has no meaning
[constr_1616]	Existence of attribute DiagnosticExtendedDataRecord.customTrigger
[constr_1617]	Existence of attribute DiagnosticFreezeFrame.customTrigger
[constr_1623]	Restriction on DiagnosticReadScalingDataByIdentifier.dataIdentifier
[constr_1624]	Existence of DiagnosticDataElement.scalingInfoSize
[constr_1633]	Existence of DiagnosticResponseOnEvent.event vs. DiagnosticResponseOnEvent.responseOnEventAction

Table E.18: Added Constraints in 4.4.0

E.5.5 Changed Constraints in 4.4.0

Number	Heading
[constr_1325]	Allowed attributes of SwDataDefProps for DiagnosticDataElement.swDataDefProps
[constr_1328]	Consistency of DiagnosticEcuInstanceProps.ecuInstance and DiagnosticServiceTable.ecuInstance

Table E.19: Changed Constraints in 4.4.0

E.5.6 Deleted Constraints in 4.4.0

Number	Heading
[constr_1471]	Existence of DiagnosticDataIdentifier.didSize
[constr_1515]	Reference from DiagnosticRoutineControl to DiagnosticAccessPermission has no meaning

Table E.20: Deleted Constraints in 4.4.0

E.6 Traceable item history of this document according to AUTOSAR Release R19-11

E.6.1 Added Specification Items in 19-11

Number	Heading
[TPS_DEXT_01147]	Support for custom service instance
[TPS_DEXT_01148]	Standardized values of DiagnosticIumprGroup.category
[TPS_DEXT_01149]	Standardized values of DiagnosticIumprDenominatorGroup.category
[TPS_DEXT_01150]	Semantics of meta-class DiagnosticControlEnableMaskBit

Table E.21: Added Specification Items in 19-11

E.6.2 Changed Specification Items in 19-11

Number	Heading
[TPS_DEXT_01060]	Applicable values for DiagnosticReadDTCInformation.category
[TPS_DEXT_01134]	Definition of a DiagnosticDataElement used in the context of a DID obtained by sender-receiver communication
[TPS_DEXT_01135]	Definition of a DiagnosticDataElement used in the context of a DID obtained by client/server communication
[TPS_DEXT_01136]	Definition of a DiagnosticDataElement used in the context of a diagnostic routine

Table E.22: Changed Specification Items in 19-11

E.6.3 Deleted Specification Items in 19-11

Number	Heading
[TPS_DEXT_01009]	Limited support for the configuration of custom diagnostic services
[TPS_DEXT_01010]	Configuration of custom diagnostic services
[TPS_DEXT_01110]	Standardized values of DiagnosticIumprGroup.category

Table E.23: Deleted Specification Items in 19-11

E.6.4 Added Constraints in 19-11

Number	Heading
[constr_1711]	Restriction of applicability of attribute typeOfFreezeFrameRecordNumeration
[constr_1721]	DiagnosticControlEnableMaskBit.bitNumber shall be unique
[constr_1722]	Relation between reference DiagnosticIOControl.dataIdentifier and attribute DiagnosticIOControl.controlEnableMaskBit
[constr_1725]	Applicability of attribute DiagnosticMemoryDestination.dtcStatusAvailabilityMask

Table E.24: Added Constraints in 19-11

E.6.5 Changed Constraints in 19-11

Number	Heading
[constr_1344]	Condition for the identification of data types of attributes DiagnosticServiceDataMapping.mappedDataElement
[constr_1349]	Value of udsDtcValue shall be unique
[constr_1354]	Existence of attribute DiagnosticTroubleCodeProps.snapshotRecordContent
[constr_1459]	Existence of attributes of DiagnosticTroubleCodeProps

Table E.25: Changed Constraints in 19-11

E.6.6 Deleted Constraints in 19-11

none

E.7 Traceable item history of this document according to AUTOSAR Release R20-11

E.7.1 Added Specification Items in R20-11

Number	Heading
[TPS_DEXT_01151]	Semantics of attribute DiagnosticEvent.associatedEventIdentification
[TPS_DEXT_01152]	Semantics of meta-class DiagnosticSecurityEventReportingModeMapping

▽

△

Number	Heading
[TPS_DEXT_01153]	Semantics of meta-class DiagnosticEventToSecurityEventMapping

Table E.26: Added Specification Items in R20-11

E.7.2 Changed Specification Items in R20-11

Number	Heading
[TPS_DEXT_01041]	Semantics of attribute DiagnosticServiceDataMapping.diagnosticDataElement
[TPS_DEXT_01142]	Standardized values for DiagnosticDemProvidedDataMapping.dataProvider

Table E.27: Changed Specification Items in R20-11

E.7.3 Deleted Specification Items in R20-11

Number	Heading
[TPS_DEXT_01052]	Existence of attribute DiagnosticServiceInstance.accessPermission
[TPS_DEXT_01061]	Supported scenarios for the definition of *access permission*
[TPS_DEXT_01062]	Existence of DiagnosticServiceClass.accessPermissionValidity in an incomplete model
[TPS_DEXT_01063]	Existence of DiagnosticServiceClass.accessPermissionValidity in a complete model
[TPS_DEXT_01076]	Identification of sub-functions of diagnostic service ControlDTCSetting

Table E.28: Deleted Specification Items in R20-11

E.7.4 Added Constraints in R20-11

Number	Heading
[constr_1745]	Indirect reference to DiagnosticCommonElement
[constr_1749]	Existence of DiagnosticInfoType.dataElement
[constr_1750]	Existence of attribute DiagnosticParameterIdentifier.pidSize
[constr_1752]	Existence of references owned by DiagnosticEnableConditionPortMapping
[constr_1753]	Existence of references owned by DiagnosticStorageConditionPortMapping

▽

△

Number	Heading
[constr_1756]	Existence of attributes DiagnosticExtendedDataRecord.trigger and update
[constr_1757]	Existence of attribute DiagnosticTroubleCodeUds.udsDtcValue
[constr_1758]	Existence of attribute DiagnosticTroubleCodeObd.obdDTCValue
[constr_1759]	Existence of references owned by DiagnosticOperationCyclePortMapping
[constr_1760]	Existence of DiagnosticExtendedDataRecord.recordElement
[constr_1761]	Existence of attribute DiagnosticConnectedIndicator.healingCycle
[constr_1762]	Existence of references owned by DiagnosticEventPortMapping
[constr_1763]	Existence of attribute DiagnosticPeriodicRate.periodicRateCategory
[constr_1766]	Existence of DiagEventDebounceCounterBased.counterJumpDownValue
[constr_1767]	Existence of DiagEventDebounceCounterBased.counterJumpUpValue
[constr_1768]	Existence of attribute DiagnosticEvent.associatedEventIdentification
[constr_1772]	Unique DiagnosticSession and DiagnosticSecurityLevel for diagnostic routines that have the same identifier
[constr_1780]	Existence of attribute DiagnosticTroubleCodeJ1939.fmi
[constr_1781]	Existence of attribute DiagnosticTroubleCodeJ1939.spn
[constr_1782]	Usage of internal data elements only for extended data records
[constr_1790]	Existence of attribute DiagnosticParameter.bitOffset
[constr_1791]	Existence of attribute DiagnosticParameter.dataElement
[constr_1792]	Existence of DiagnosticDataIdentifier.dataElement
[constr_1793]	Existence of attribute DiagnosticAbstractDataIdentifier.id
[constr_1794]	Existence of attribute DiagnosticProtocol.priority
[constr_1795]	Existence of attribute DiagnosticProtocol.protocolKind
[constr_1796]	Existence of attribute DiagnosticServiceTable.serviceInstance
[constr_1797]	Existence of attribute DiagnosticServiceTable.protocolKind
[constr_1798]	Existence of DiagnosticServiceInstance.serviceClass
[constr_1799]	Existence of DiagnosticEnvironmentalCondition.formula
[constr_1800]	Existence of DiagnosticEnvConditionFormula.op
[constr_1801]	Existence of DiagnosticEnvCompareCondition.compareType
[constr_1802]	Existence of DiagnosticEnvDataCondition.compareValue
[constr_1803]	Existence of DiagnosticEnvDataCondition.dataElement
[constr_1804]	Existence of DiagnosticEnvModeCondition.modeElement
[constr_1805]	Existence of DiagnosticEnvSwcModeElement.mode
[constr_1806]	Existence of DiagnosticEnvBswModeElement.mode
[constr_1807]	Existence of reference DiagnosticDataByIdentifier.dataIdentifier
[constr_1808]	Existence of reference DiagnosticDynamicallyDefineData-Identifier.dataIdentifier
[constr_1810]	Existence of aggregation DiagnosticReadDataByPeriodic-IDClass.periodicRate
[constr_1811]	Existence of attribute DiagnosticReadDataByPeriodicIDClass.max-PeriodicDidToRead

▽

△

Number	Heading
[constr_1812]	Existence of attribute DiagnosticReadDataByPeriodicIDClass.scheduler-MaxNumber
[constr_1813]	Existence of reference DiagnosticEventWindow.eventWindowTime
[constr_1814]	Existence of reference DiagnosticEventWindow.storageStateEvaluation
[constr_1815]	Existence of attribute DiagnosticRoutine.id
[constr_1816]	Existence of attribute DiagnosticSecurityAccess.requestSeedId
[constr_1817]	Existence of attribute DiagnosticSecurityAccess.securityLevel
[constr_1818]	Existence of reference DiagnosticSessionControl.diagnosticSession
[constr_1819]	Existence of attribute DiagnosticParameterIdentifier.id
[constr_1820]	Existence of reference DiagnosticRequestCurrentPowertrainData.pid
[constr_1821]	Existence of reference DiagnosticRequestPowertrainFreezeFrame-Data.freezeFrame
[constr_1822]	Existence of reference DiagnosticRequestControlOfOnBoard-Device.testId
[constr_1823]	Existence of attribute DiagnosticTestRoutineIdentifier.id
[constr_1824]	Existence of attribute DiagnosticTestRoutineIdentifier.requestData-Size
[constr_1825]	Existence of attribute DiagnosticTestRoutineIdentifier.responseData-Size
[constr_1826]	Existence of reference DiagnosticRequestVehicleInfo.infoType
[constr_1827]	Existence of attribute DiagnosticInfoType.id
[constr_1828]	Existence of referenced from DiagnosticServiceDataMapping
[constr_1829]	Existence of reference DiagnosticConnectedIndicator.indicator
[constr_1830]	Existence of DiagnosticTroubleCodeGroup.groupNumber
[constr_1831]	Existence of DiagnosticTroubleCodeProps.priority
[constr_1832]	Existence of DiagnosticExtendedDataRecord.recordNumber
[constr_1833]	Existence of DiagnosticFreezeFrame.trigger
[constr_1834]	Existence of DiagnosticCondition.initValue
[constr_1835]	Existence of DiagEventDebounceCounterBased.counterDecrementStep-Size
[constr_1836]	Existence of DiagEventDebounceCounterBased.counterIncrementStep-Size
[constr_1837]	Existence of DiagEventDebounceCounterBased.counterFailedThreshold
[constr_1838]	Existence of DiagEventDebounceCounterBased.counterPassedThreshold
[constr_1839]	Existence of attribute DiagEventDebounceTimeBased.timeFailedThreshold
[constr_1840]	Existence of attribute DiagEventDebounceTimeBased.timePassedThreshold
[constr_1841]	Existence of DiagnosticEnableConditionGroup.enableCondition
[constr_1842]	Existence of DiagnosticStorageConditionGroup.storageCondition
[constr_1843]	Existence of reference DiagnosticEventPortMapping.diagnosticEvent
[constr_1844]	Existence of reference DiagnosticOperationCyclePortMapping.operation-Cycle

▽

△

Number	Heading
[constr_1845]	Existence of reference DiagnosticEnableConditionPortMapping.enableCondition
[constr_1846]	Existence of reference DiagnosticStorageConditionPortMapping.diagnosticStorageCondition
[constr_1847]	Existence of reference DiagnosticDemProvidedDataMapping.dataElement
[constr_1848]	Existence of attribute DiagnosticAging.agingCycle
[constr_1849]	Existence of attribute DiagnosticAging.threshold
[constr_1850]	Existence of aggregation DiagnosticTestResult.testIdentifier
[constr_1851]	Existence of reference DiagnosticTestResult.monitoredIdentifier
[constr_1852]	Existence of attribute DiagnosticEcuInstanceProps.obdSupport
[constr_1853]	Existence of attribute DiagnosticIumprGroup.iumprGroupIdentifier
[constr_1854]	Existence of attribute DiagnosticIumprGroupIdentifier.groupId
[constr_1855]	Existence of attribute DiagnosticFunctionIdentifierInhibit.inhibitionMask
[constr_1856]	Existence of attribute DiagnosticJ1939Spn.spn
[constr_1857]	Existence of the reference DiagnosticEventToTroubleCodeJ1939Mapping.diagnosticEvent
[constr_1858]	Existence of the attribute DiagnosticEventToTroubleCodeJ1939Mapping.troubleCodeJ1939
[constr_1859]	Usage of DiagnosticRecordTriggerEnum.testFailedThisOperationCycle
[constr_10024]	Existence of reference in the role DiagnosticSecurityEventReportingModeMapping.dataElement
[constr_10025]	Existence of reference in the role DiagnosticSecurityEventReportingModeMapping.securityEvent
[constr_10026]	Existence of reference in the role DiagnosticEventToSecurityEventMapping.diagnosticEvent
[constr_10027]	Existence of reference in the role DiagnosticEventToSecurityEventMapping.securityEventProps

Table E.29: Added Constraints in R20-11

E.7.5 Changed Constraints in R20-11

none

E.7.6 Deleted Constraints in R20-11

Number	Heading
[constr_1360]	Usage of DiagEventDebounceMonitorInternal is not supported in the context of DiagnosticDebounceAlgorithmProps

Table E.30: Deleted Constraints in R20-11

E.8 Traceable item history of this document according to AUTOSAR Release R21-11

E.8.1 Added Specification Items in R21-11

Number	Heading
[TPS_DEXT_01154]	Semantics of meta-class `DiagnosticAuthRole`
[TPS_DEXT_01155]	Semantics of `DiagnosticMemoryDestinationUserDefined`. `authenticationRole`
[TPS_DEXT_01156]	Further definition of the semantics of `DiagnosticIumpr`
[TPS_DEXT_01157]	Semantics of `DiagnosticIumprToFunctionIdentifierMapping`
[TPS_DEXT_01158]	Sub-Functions for diagnostic service `Authentication`
[TPS_DEXT_01159]	Supported authentication methods for diagnostic service `Authentication`
[TPS_DEXT_01160]	Semantics of the reference `DiagnosticMapping.providerSoftwareCluster`
[TPS_DEXT_01161]	Mapping between a `DiagnosticEvent` and a `CpSoftwareClusterResource`
[TPS_DEXT_01162]	Mapping between a `DiagnosticDataElement` and a `CpSoftwareClusterResource`
[TPS_DEXT_01163]	Mapping between a `DiagnosticRoutineSubfunction` and a `CpSoftwareClusterResource`
[TPS_DEXT_01164]	Mapping between a `DiagnosticFunctionIdentifier` and a `CpSoftwareClusterResource`

Table E.31: Added Specification Items in R21-11

E.8.2 Changed Specification Items in R21-11

Number	Heading
[TPS_DEXT_01045]	Supported diagnostic services
[TPS_DEXT_01060]	Applicable values for `DiagnosticReadDTCInformation.category`
[TPS_DEXT_01070]	Description of textually semi-formal formulated pre- and run-conditions for the validity of `DiagnosticAccessPermission`
[TPS_DEXT_01091]	Relation between a `DiagnosticServiceTable` and one or more `DiagnosticConnection`s
[TPS_DEXT_01093]	Definition of legislative freeze frame for WWWH-OBD
[TPS_DEXT_01111]	Legislative freeze frame for the OBD-II case
[TPS_DEXT_01112]	Definition of a diagnostic trouble code for the implementation of WWH-OBD
[TPS_DEXT_01142]	Standardized values for `DiagnosticDemProvidedDataMapping.dataProvider`
[TPS_DEXT_03012]	Three kinds of DTCs

Table E.32: Changed Specification Items in R21-11

E.8.3 Deleted Specification Items in R21-11

Number	Heading
[TPS_DEXT_01033]	Semantics of triggers in the context of a DiagnosticResponseOnEvent
[TPS_DEXT_01049]	Consistency of DiagnosticServiceSwMapping with respect to routine IDs
[TPS_DEXT_01050]	Consistency of DiagnosticServiceSwMapping with respect to data IDs
[TPS_DEXT_01051]	Consistency of DiagnosticServiceSwMapping with respect to data IDs
[TPS_DEXT_01054]	Existence of DiagnosticDataByIdentifier.dataIdentifier

Table E.33: Deleted Specification Items in R21-11

E.8.4 Added Constraints in R21-11

Number	Heading
[constr_10038]	Restriction for the usage of DiagnosticAccessPermission. authenticationRole
[constr_10042]	Existence of attribute DiagnosticCommonProps.defaultEndianness
[constr_10043]	Existence of attribute DiagnosticCommonProps.resetConfirmedBitOnOverflow
[constr_10044]	Existence of attribute DiagnosticCommonProps.occurrenceCounterProcessing
[constr_10045]	Existence of attribute DiagnosticSecurityAccessClass.securityDelayTimeOnBoot
[constr_10084]	Existence of DiagnosticIumprToFunctionIdentifierMapping.iumpr
[constr_10085]	Existence of DiagnosticIumprToFunctionIdentifierMapping.functionIdentifier
[constr_10088]	Relation between event and DTC without event combination
[constr_10089]	Existence of attribute DiagnosticCommonProps.eventCombinationReportingBehavior
[constr_10091]	Mandatory subfunction of diagnostic service Authentication

Table E.34: Added Constraints in R21-11

E.8.5 Changed Constraints in R21-11

Number	Heading
[constr_1349]	Value of udsDtcValue shall be unique
[constr_1457]	Service-only DTCs shall refer to a common memory section
[constr_1458]	Reference to DiagnosticMemoryDestination
[constr_1459]	Existence of attributes of DiagnosticTroubleCodeProps

▽

△

Number	Heading
[constr_1509]	extendedDataRecord.recordNumber shall be unique within primary fault memory
[constr_1511]	extendedDataRecord.recordNumber shall be unique within user-defined fault memory
[constr_1512]	freezeFrame.recordNumber shall be unique within primary fault memory
[constr_1514]	freezeFrame.recordNumber shall be unique within user-defined fault memory

Table E.35: Changed Constraints in R21-11

E.8.6 Deleted Constraints in R21-11

Number	Heading
[constr_1365]	Multiplicity of DiagnosticResponseOnEvent.event
[constr_1366]	Event ID in the context of diagnostic service ResponseOnEvent shall be unique
[constr_1376]	Multiplicity of reference DiagnosticTroubleCodeProps.memoryDestination
[constr_1377]	Existence of reference DiagnosticTroubleCodeProps.memoryDestination
[constr_1380]	Existence of DiagnosticMemoryDestinationMirror
[constr_1510]	extendedDataRecord.recordNumber shall be unique within mirror fault memory
[constr_1513]	freezeFrame.recordNumber shall be unique within mirror fault memory
[constr_1633]	Existence of DiagnosticResponseOnEvent.event vs. DiagnosticResponseOnEvent.responseOnEventAction
[constr_1711]	Restriction of applicability of attribute typeOfFreezeFrameRecordNumeration
[constr_1725]	Applicability of attribute DiagnosticMemoryDestination.dtcStatusAvailabilityMask
[constr_1813]	Existence of reference DiagnosticEventWindow.eventWindowTime
[constr_1814]	Existence of reference DiagnosticEventWindow.storageStateEvaluation

Table E.36: Deleted Constraints in R21-11

E.9 Traceable item history of this document according to AUTOSAR Release R22-11

E.9.1 Added Specification Items in R22-11

Number	Heading
[TPS_DEXT_01165]	Semantics of DiagnosticEnvDataElementCondition
[TPS_DEXT_01166]	Usage of DiagnosticEnvDataElementCondition in top-down and bottom-up scenarios

▽

△

Number	Heading
[TPS_DEXT_01167]	Relevance of CompuMethod and DataConstr for the definition of DiagnosticEnvDataElementCondition
[TPS_DEXT_01168]	Allowed values of compareType in the context of a DiagnosticEnvDataCondition
[TPS_DEXT_01169]	Role of meta-class DiagnosticParameter
[TPS_DEXT_01170]	Semantics of meta-class DiagnosticAbstractParameter
[TPS_DEXT_01171]	Size of a DiagnosticAbstractParameter
[TPS_DEXT_01172]	Semantics of attribute DiagnosticAbstractParameter.parameterSize
[TPS_DEXT_01173]	Start position of a DiagnosticAbstractParameter
[TPS_DEXT_01174]	Content of a DiagnosticParameter
[TPS_DEXT_01175]	Nested definition of a DiagnosticParameter
[TPS_DEXT_01176]	DiagnosticParameter as a reference target
[TPS_DEXT_01177]	Utility of DiagnosticServiceMappingDiagTarget
[TPS_DEXT_01178]	Semantics of DiagnosticServiceDataMapping.parameterElementAccess
[TPS_DEXT_01179]	Existence of reference to DiagnosticParameterIdent in the role diagnosticParameter
[TPS_DEXT_01180]	Existence of reference to DiagnosticParameterElement in the role diagnosticParameter
[TPS_DEXT_01181]	Existence of references to subclasses of DiagnosticServiceMappingDiagTarget
[TPS_DEXT_01182]	Use case for the existence of reference DiagnosticServiceSwMapping.accessedDataPrototype

Table E.37: Added Specification Items in R22-11

E.9.2 Changed Specification Items in R22-11

Number	Heading
[TPS_DEXT_01139]	Semantics of the references from DiagnosticAccessPermission
[TPS_DEXT_01149]	Standardized values of DiagnosticIumprDenominatorGroup.category
[TPS_DEXT_03012]	Three kinds of DTCs

Table E.38: Changed Specification Items in R22-11

E.9.3 Deleted Specification Items in R22-11

Number	Heading
[TPS_DEXT_01155]	Semantics of DiagnosticMemoryDestinationUserDefined. authenticationRole

Table E.39: Deleted Specification Items in R22-11

E.9.4 Added Constraints in R22-11

Number	Heading
[constr_10100]	Existence of DiagnosticRoutineControl.routine
[constr_10115]	Existence of attributes of DiagnosticEnvDataElementCondition if the reference in the role dataPrototype exists
[constr_10116]	Existence of attributes of DiagnosticEnvDataElementCondition if the reference in the role dataPrototype does not exist
[constr_10117]	Existence of attributes of DiagnosticEnvDataElementCondition.swDataDefProps
[constr_10122]	Existence of attribute DiagnosticComControlSubNodeChannel.subNodeChannel
[constr_10364]	Usage of DiagnosticRecordTriggerEnum.testPassed
[constr_10368]	Restriction regarding the reference DiagnosticDataIdentifierSet. dataIdentifier
[constr_10369]	Existence of attributes of DiagnosticParameterElement depending on the value of attribute category
[constr_10370]	Restriction regarding the role DiagnosticParameterIdentifier.dataElement
[constr_10371]	Restriction regarding the aggregation of DiagnosticParameter in the role DiagnosticExtendedDataRecord.recordElement

Table E.40: Added Constraints in R22-11

E.9.5 Changed Constraints in R22-11

Number	Heading
[constr_1470]	Value of DiagnosticAbstractParameter.bitOffset
[constr_1624]	Existence of DiagnosticDataElement.scalingInfoSize
[constr_1782]	Usage of internal data elements only for extended data records
[constr_1790]	Existence of attribute DiagnosticAbstractParameter.bitOffset
[constr_1791]	Existence of attribute dataElement vs. parameterSize of meta-class DiagnosticParameter
[constr_1819]	Existence of attribute DiagnosticParameterIdentifier.id



△

Number	Heading
[constr_1820]	Existence of reference DiagnosticRequestCurrentPowertrainData.pid
[constr_1821]	Existence of reference DiagnosticRequestPowertrainFreezeFrameData.freezeFrame
[constr_1822]	Existence of reference DiagnosticRequestControlOfOnBoardDevice.testId
[constr_1823]	Existence of attribute DiagnosticTestRoutineIdentifier.id
[constr_1824]	Existence of attribute DiagnosticTestRoutineIdentifier.requestDataSize
[constr_1825]	Existence of attribute DiagnosticTestRoutineIdentifier.responseDataSize
[constr_1826]	Existence of reference DiagnosticRequestVehicleInfo.infoType
[constr_1827]	Existence of attribute DiagnosticInfoType.id
[constr_1828]	Existence of referenced from DiagnosticServiceDataMapping

Table E.41: Changed Constraints in R22-11

E.9.6 Deleted Constraints in R22-11

Number	Heading
[constr_1344]	Condition for the identification of data types of attributes DiagnosticServiceDataMapping.mappedDataElement
[constr_1465]	Allowed values of compareType in the context of a DiagnosticEnvDataCondition

Table E.42: Deleted Constraints in R22-11

E.10 Traceable item history of this document according to AUTOSAR Release R23-11

E.10.1 Added Specification Items in R23-11

Number	Heading
[TPS_DEXT_01183]	Existence of attributes of DiagnosticTroubleCodeProps
[TPS_DEXT_01184]	Use cases for reading data from the application
[TPS_DEXT_01185]	Use cases for writing data to the application
[TPS_DEXT_01186]	Impact of the DiagnosticComControlClass on the state management for VLANs

▽

△

Number	Heading
[TPS_DEXT_01187]	References from DiagnosticSecureCodingMapping to DiagnosticRoutine
[TPS_DEXT_01188]	Attribute DiagnosticAccessPermission.authenticationEnabled does not exist
[TPS_DEXT_01189]	Attribute DiagnosticAccessPermission.authenticationEnabled exists
[TPS_DEXT_01190]	Attribute DiagnosticAccessPermission.authenticationEnabled exists, but DiagnosticAuthRoleProxy.authenticationRole does not exist
[TPS_DEXT_01191]	Attribute DiagnosticAccessPermission.authenticationEnabled and DiagnosticAuthRoleProxy.authenticationRole exist
[TPS_DEXT_01192]	Semantics of attribute DiagnosticAuthTransmitCertificateEvaluation.evaluationId
[TPS_DEXT_01193]	Semantics of attribute DiagnosticAuthTransmitCertificateEvaluation.function
[TPS_DEXT_01194]	Standardized value of attribute DiagnosticAuthTransmitCertificateEvaluation.function
[TPS_DEXT_01195]	Semantics of the DiagnosticAuthTransmitCertificateMapping

Table E.43: Added Specification Items in R23-11

E.10.2 Changed Specification Items in R23-11

Number	Heading
[TPS_DEXT_01011]	Semantics of DiagnosticSession.id
[TPS_DEXT_01045]	Supported diagnostic services in the scope of this document
[TPS_DEXT_01139]	Semantics of the references from DiagnosticAccessPermission
[TPS_DEXT_01142]	Standardized values for attribute DiagnosticDemProvidedDataMapping.dataProvider
[TPS_DEXT_01149]	Standardized values of DiagnosticIumprDenominatorGroup.category

Table E.44: Changed Specification Items in R23-11

E.10.3 Deleted Specification Items in R23-11

none

E.10.4 Added Constraints in R23-11

Number	Heading
[constr_10412]	Existence of attribute DiagnosticTestIdentifier.id
[constr_10413]	Existence of attribute DiagnosticTestIdentifier.uasId
[constr_10414]	Existence of attribute DiagnosticMeasurementIdentifier.obdMid
[constr_10418]	Existence of attribute DiagnosticDebounceAlgorithmProps.debounceAlgorithm
[constr_10419]	Existence of the attribute DiagnosticCommonProps.resetPendingBitOnOverflow
[constr_10421]	Existence of attribute DiagnosticMemoryDestination.dtcStatusAvailabilityMask
[constr_10422]	Existence of attribute DiagnosticMemoryDestination.eventDisplacementStrategy
[constr_10423]	Existence of attribute DiagnosticMemoryDestination.maxNumberOfEventEntries
[constr_10440]	Restriction for the minimum value of attribute DiagnosticSessionControlClass.s3ServerTimeout

Table E.45: Added Constraints in R23-11

E.10.5 Changed Constraints in R23-11

Number	Heading
[constr_1329]	Existence of concrete sub-classes of DiagnosticServiceClass in the context created by a DiagnosticContributionSet
[constr_1333]	Existence of attributes of meta-class DiagnosticMemoryIdentifier
[constr_1552]	Meta-class DiagnosticDataIdentifier referenced by DiagnosticDataIdentifierSet
[constr_10038]	Restriction for the usage of DiagnosticAccessPermission.authenticationEnabled
[constr_10371]	DiagnosticDataElements owned by a DiagnosticExtendedDataRecord

Table E.46: Changed Constraints in R23-11

E.10.6 Deleted Constraints in R23-11

Number	Heading
[constr_1459]	Existence of attributes of DiagnosticTroubleCodeProps

Table E.47: Deleted Constraints in R23-11

E.10.7 Added Advisories in R23-11

none

E.10.8 Changed Advisories in R23-11

none

E.10.9 Deleted Advisories in R23-11

none

E.11 Traceable item history of this document according to AUTOSAR Release R24-11

E.11.1 Added Specification Items in R24-11

none

E.11.2 Changed Specification Items in R24-11

Number	Heading
[TPS_DEXT_01045]	Supported diagnostic services in the scope of this document
[TPS_DEXT_01142]	Standardized values for attribute `DiagnosticDemProvidedDataMapping.dataProvider`
[TPS_DEXT_01151]	Semantics of attribute `DiagnosticEvent.associatedEventIdentification`
[TPS_DEXT_03012]	Three kinds of DTCs

Table E.48: Changed Specification Items in R24-11

E.11.3 Deleted Specification Items in R24-11

Number	Heading
[TPS_DEXT_01183]	Existence of attributes of `DiagnosticTroubleCodeProps`

Table E.49: Deleted Specification Items in R24-11

E.11.4 Added Constraints in R24-11

Number	Heading
[constr_10522]	OBD trouble code shall only be placed in primary fault memory
[constr_10523]	Existence of role `DiagnosticTroubleCodeUdsToTroubleCodeObdMapping.troubleCodeUds`
[constr_10524]	Existence of role `DiagnosticTroubleCodeUdsToTroubleCodeObdMapping.troubleCodeObd`
[constr_10545]	Existence of `DiagnosticParameterIdentifier.dataElement`.
[constr_10573]	Existence of attribute `DiagnosticServiceTable.diagnosticServiceInstance`
[constr_10577]	Existence of `DiagnosticResponseOnEventClass.storeEventEnabled`

Table E.50: Added Constraints in R24-11

E.11.5 Changed Constraints in R24-11

Number	Heading
[constr_10088]	Relation between event and DTC without event combination
[constr_1349]	Value of `udsDtcValue` shall be unique
[constr_1768]	Existence of attribute `DiagnosticEvent.associatedEventIdentification`

Table E.51: Changed Constraints in R24-11

E.11.6 Deleted Constraints in R24-11

Number	Heading
[constr_1796]	Existence of attribute `DiagnosticServiceTable.serviceInstance`

Table E.52: Deleted Constraints in R24-11

E.11.7 Added Advisories in R24-11

none

E.11.8 Changed Advisories in R24-11

none

E.11.9 Deleted Advisories in R24-11

none

E.12 Traceable item history of this document according to AUTOSAR Release R25-11

E.12.1 Added Specification Items in R25-11

Number	Heading
[TPS_DEXT_01196]	Reference from DiagnosticRoutineControl to DiagnosticAccessPermission has no meaning
[TPS_DEXT_01197]	Semantics of the DiagnosticEdrSenderPortMapping
[TPS_DEXT_01198]	Semantics of the DiagnosticEdrServerPortMapping
[TPS_DEXT_01199]	Semantics of meta-class ISignalPortToDiagnosticEventMapping
[TPS_DEXT_01200]	Semantics of the reference in the role ISignalPortToDiagnosticEventMapping.iSignalTriggering

Table E.53: Added Specification Items in R25-11

E.12.2 Changed Specification Items in R25-11

Number	Heading
[TPS_DEXT_01134]	Definition of a DiagnosticDataElement used in the context of a DID obtained by sender-receiver communication
[TPS_DEXT_01135]	Definition of a DiagnosticDataElement used in the context of a DID obtained by client/server communication
[TPS_DEXT_01136]	Definition of a DiagnosticDataElement used in the context of a diagnostic routine
[TPS_DEXT_01151]	Semantics of attribute DiagnosticEvent.associatedEventIdentification
[TPS_DEXT_03008]	Semantics of DiagnosticExtendedDataRecord
[TPS_DEXT_03020]	Semantics of DiagnosticEdrDataProviderMapping

Table E.54: Changed Specification Items in R25-11

E.12.3 Deleted Specification Items in R25-11

none

E.12.4 Added Constraints in R25-11

Number	Heading
[constr_10642]	Existence of DiagnosticEventPortMapping.diagnosticEvent for one specific DiagnosticEvent
[constr_10645]	Value of DiagnosticEvent.confirmationThreshold for an n:1 mapping to DiagnosticTroubleCodeUds
[constr_10646]	Value of attributes of DiagnosticConnectedIndicator for an n:1 mapping to DiagnosticTroubleCodeUds where all affected DiagnosticEvents transitively refer to the identical DiagnosticIndicator
[constr_10647]	In an n:1 mapping between DiagnosticEvent and DiagnosticTroubleCodeUds, all DiagnosticEvents shall be mapped to the same DiagnosticOperationCycle
[constr_10665]	Existence of attribute DiagnosticCommonProps.authenticationTimeout
[constr_10676]	Allowed attribute of attribute DiagnosticExtendedDataRecordElement.swDataDefProps
[constr_10677]	Allowed values for attribute DiagnosticExtendedDataRecordElement.swDataDefProps.baseType.baseTypeSize
[constr_10678]	Existence of DiagnosticEdrSenderPortMapping.dataPrototype
[constr_10679]	Existence of DiagnosticEdrSenderPortMapping.recordElement
[constr_10680]	Existence of DiagnosticEdrServerPortMapping.swcServiceDependencyInSystem vs. bswServiceDependency
[constr_10681]	Existence of DiagnosticEdrServerPortMapping.recordElement
[constr_10687]	Existence of reference ISignalPortToDiagnosticEventMapping.iSignalTriggering
[constr_10688]	Existence of reference ISignalPortToDiagnosticEventMapping.iSignalPort
[constr_10689]	Restriction for attribute communicationDirection of an ISignalPort that is referenced by ISignalPortToDiagnosticEventMapping

Table E.55: Added Constraints in R25-11

E.12.5 Changed Constraints in R25-11

Number	Heading
[constr_10044]	Existence of attribute DiagnosticCommonProps.occurrenceCounterProcessing
[constr_10089]	Existence of attribute DiagnosticCommonProps.eventCombinationReportingBehavior
[constr_10368]	Restriction regarding the reference DiagnosticDataIdentifierSet.diagnosticDataIdentifier
[constr_1435]	Debouncing in the presence of a DiagnosticEventPortMapping
[constr_1756]	Existence of attributes DiagnosticExtendedDataRecord.trigger and update

▽

△

Number	Heading
[constr_1760]	Existence of DiagnosticExtendedDataRecord.element
[constr_1768]	Existence of attribute DiagnosticEvent.associatedEventIdentification
[constr_1847]	Existence of reference DiagnosticEdrDataProviderMapping.extendedDataRecordElement

Table E.56: Changed Constraints in R25-11

E.12.6 Deleted Constraints in R25-11

Number	Heading
[constr_10371]	DiagnosticDataElements owned by a DiagnosticExtendedDataRecord
[constr_1612]	Reference from DiagnosticRoutineControl to DiagnosticAccessPermission has no meaning
[constr_1761]	Existence of attribute DiagnosticConnectedIndicator.healingCycle
[constr_1782]	Usage of internal data elements only for extended data records

Table E.57: Deleted Constraints in R25-11

E.12.7 Added Advisories in R25-11

none

E.12.8 Changed Advisories in R25-11

none

E.12.9 Deleted Advisories in R25-11

none

E.13 Traceable item history of this document according to AUTOSAR Release R26-11

E.13.1 Added Specification Items in R26-11

Number	Heading
[TPS_DEXT_01201]	Semantics of meta-class DiagnosticTroubleCodeUdsToTroubleCodeObdMapping
[TPS_DEXT_01202]	Applicability of DiagnosticReadDataByPeriodicID
[TPS_DEXT_01203]	Existence of DiagnosticExtendedDataRecordElement.swDataDefProps

Table E.58: Added Specification Items in R26-11

E.13.2 Changed Specification Items in R26-11

Number	Heading
[TPS_DEXT_01142]	Standardized values for attribute DiagnosticEdrDataProviderMapping.dataProvider
[TPS_DEXT_01160]	Semantics of the reference in the role DiagnosticMapping.providerSoftwareCluster
[TPS_DEXT_01161]	Mapping between a DiagnosticEvent and a CpSoftwareClusterResource
[TPS_DEXT_01162]	Creation of a mapping between a DiagnosticDataElement and a CpSoftwareClusterResource
[TPS_DEXT_01163]	Mapping between a DiagnosticRoutineSubfunction and a CpSoftwareClusterResource
[TPS_DEXT_01164]	Mapping between a DiagnosticFunctionIdentifier and a CpSoftwareClusterResource
[TPS_DEXT_03012]	Three kinds of DTCs

Table E.59: Changed Specification Items in R26-11

E.13.3 Deleted Specification Items in R26-11

Number	Heading
[TPS_DEXT_01094]	Semantics of meta-class DiagnosticTroubleCodeUdsToTroubleCodeObdMapping
[TPS_DEXT_01152]	Semantics of meta-class DiagnosticSecurityEventReportingModeMapping

Table E.60: Deleted Specification Items in R26-11

E.13.4 Added Constraints in R26-11

Number	Heading
[constr_10750]	Existence of `DiagnosticTroubleCodeUdsToTroubleCodeObdMapping`
[constr_10751]	Existence of reference to `EventObdReadinessGroup`
[constr_10752]	Existence of `DiagnosticReadDataByPeriodicID`

Table E.61: Added Constraints in R26-11

E.13.5 Changed Constraints in R26-11

none

E.13.6 Deleted Constraints in R26-11

Number	Heading
[constr_10024]	Existence of reference in the role `DiagnosticSecurityEventReportingModeMapping.dataElement`
[constr_10025]	Existence of reference in the role `DiagnosticSecurityEventReportingModeMapping.securityEvent`

Table E.62: Deleted Constraints in R26-11

E.13.7 Added Advisories in R26-11

none

E.13.8 Changed Advisories in R26-11

none

E.13.9 Deleted Advisories in R26-11

none

F Modeling of InstanceRef

F.1 Introduction

The existence of so-called InstanceRefs is a direct consequence to the usage of the type-prototype pattern for modeling within AUTOSAR. When referencing a prototype it is also necessary to include a reference to the prototypes typed by their corresponding types that in turn aggregate further prototypes to set up the context.

In other words, InstanceRefs are representing **structured references** that, on the one hand, consist of references to context prototypes (indicated by a subsetting or redefinition of atpContextElement) and finally a reference to the applicable target prototype (indicated by a redefinition of atpTarget).

Note that it is not uncommon to have more than a single context in the modeling of particular InstanceRefs.

For the reader of specifications, the modeling of InstanceRefs manifests as a UML dependency stereotyped ≪instanceRef≫ drawn from one meta-class to another.

This is a simplified indication that the source of the dependency implements an InstanceRef to the meta-class at the target of the dependency. Again, in most cases this is everything a reader needs to understand in order to figure out the modeling.

The formal modeling of InstanceRefs is done by creating subclasses of the abstract meta-class AtpInstanceRef.

Wherever a more detailed understanding of the modeling is advised in the context of the specific chapter of this document, the modeling of a specific subclasses of AtpInstanceRef is explained directly in the context of the corresponding chapter.

In all other cases, a deeper understanding of the modeling of particular subclasses of AtpInstanceRefs can be obtained from reading this chapter.

Class tables included in this chapter are not fully filled out in the sense that most of the notes inside the class tables are missing.

The **primary** purpose of these class tables is to **provide information about the intended order** in which InstanceRefs are **serialized in M1 AUTOSAR models**.

In particular, the information about the order in serialized M1 models can be obtained from the value of the tag xml.sequenceOffset of each attribute of an InstanceRef meta-class.

For more information about the general concept of modeling AtpInstanceRef (e.g. the conceptual background of redefining or subsetting an association from a subclass of AtpInstanceRef to other meta-classes) please refer to the [15, TPS Generic Structure Template].

F.2 Modeling

Class	**DataPrototypeInSystemInstanceRef**			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	DiagnosticEdrSenderPortMapping.dataPrototype, DiagnosticEnvDataElementCondition.dataPrototype, DiagnosticServiceDataMapping.mappedDataElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	System	0..1	ref	This represents the base of the InstanceRef **Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
contextComponent (ordered)	SwComponentPrototype	*	ref	**Tags:** xml.sequenceOffset=30
contextDataPrototype (ordered)	ApplicationCompositeElementDataPrototype	*	ref	**Tags:** xml.sequenceOffset=50
contextPort	PortPrototype	0..1	ref	This represents the PortPrototype that is contained in the InstanceRef. **Tags:** xml.sequenceOffset=40
contextRootComposition	RootSwCompositionPrototype	0..1	ref	**Tags:** xml.sequenceOffset=20
rootDataPrototype	AutosarDataPrototype	0..1	ref	**Tags:** xml.sequenceOffset=45
targetDataPrototype	DataPrototype	0..1	ref	This represents the target of the InstanceRef **Tags:** xml.sequenceOffset=60

Table F.1: DataPrototypeInSystemInstanceRef

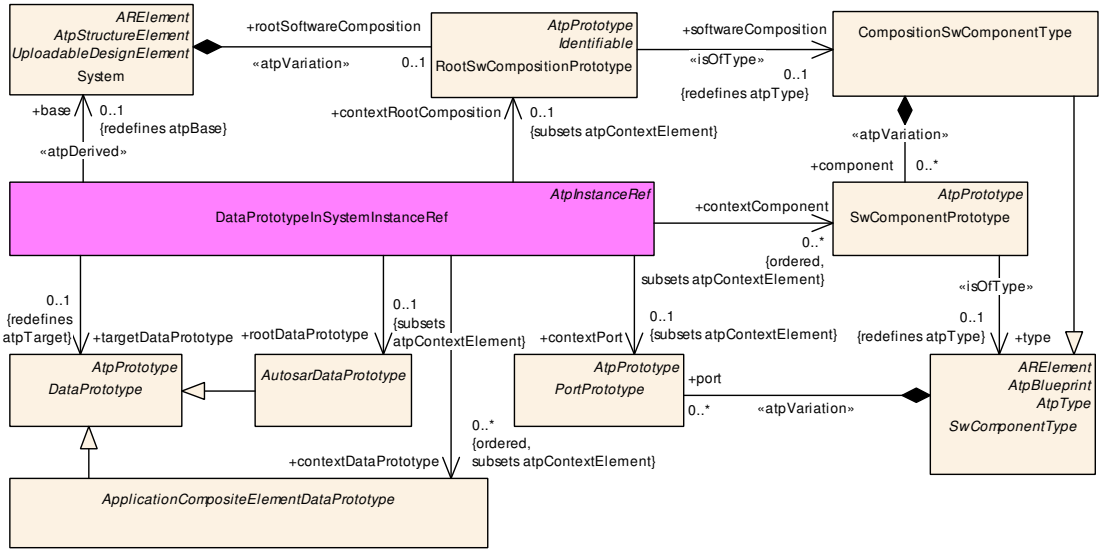

Figure F.1: Modeling of `DataPrototypeInSystemInstanceRef`

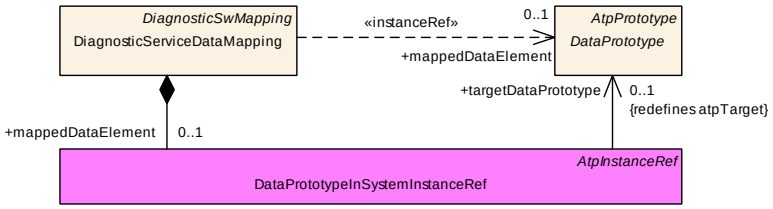

Figure F.2: Modeling of `DiagnosticServiceDataMapping`

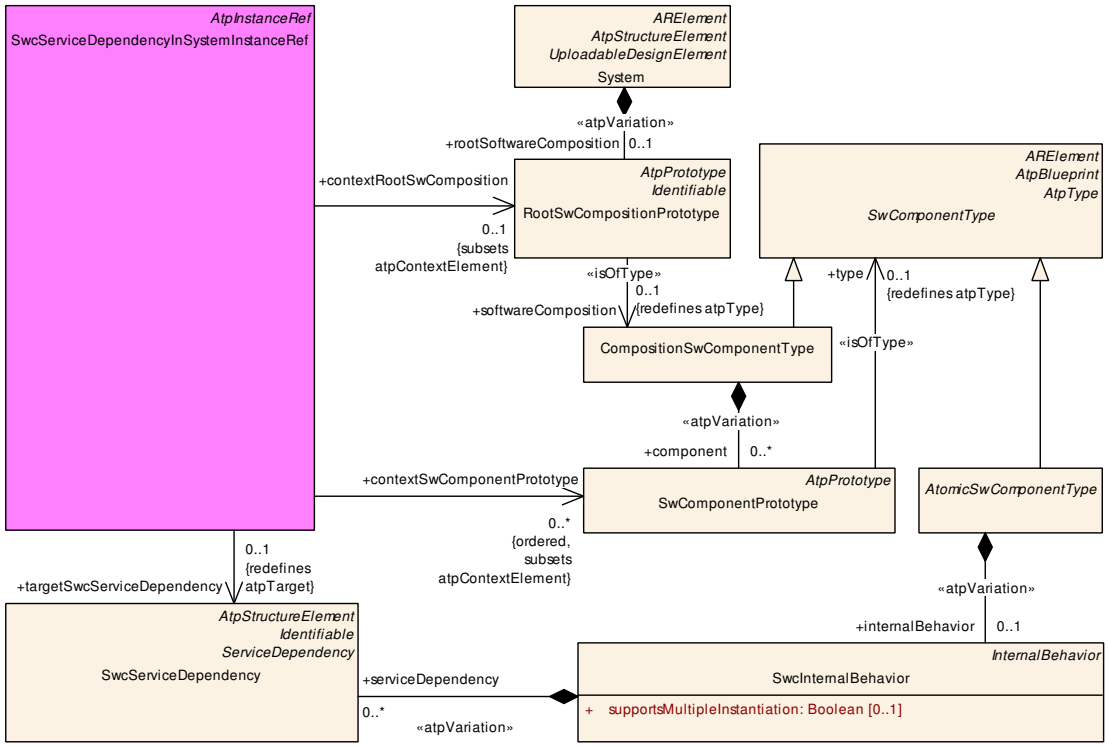

Figure F.3: Modeling of `SwcServiceDependencyInSystemInstanceRef`

Class	**SwcServiceDependencyInSystemInstanceRef**			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	DiagnosticEdrServerPortMapping.swcServiceDependencyInSystem, DiagnosticEnableConditionPortMapping.swcServiceDependencyInSystem, DiagnosticEventPortMapping.swcServiceDependencyInSystem, DiagnosticFimFunctionMapping.mappedSwcServiceDependency, DiagnosticOperationCyclePortMapping.swcServiceDependencyInSystem, DiagnosticServiceSwMapping.mappedSwcServiceDependencyInSystem, DiagnosticStorageConditionPortMapping.swcServiceDependencyInSystem			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
contextRootSwComposition	RootSwCompositionPrototype	0..1	ref	
contextSwComponentPrototype (ordered)	SwComponentPrototype	*	ref	

△

Class	**SwcServiceDependencyInSystemInstanceRef**			
targetSwc Service Dependency	SwcService Dependency	0..1	ref	

Table F.2: SwcServiceDependencyInSystemInstanceRef

Class	**PModeInSystemInstanceRef**			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	DiagnosticEnvSwcModeElement.mode			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	System	0..1	ref	**Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
context Component (ordered)	SwComponent Prototype	*	ref	**Tags:** xml.sequenceOffset=30
context Composition	RootSwComposition Prototype	0..1	ref	**Tags:** xml.sequenceOffset=20
contextMode Declaration Group	ModeDeclarationGroup Prototype	0..1	ref	**Tags:** xml.sequenceOffset=50
contextPPort	AbstractProvidedPort Prototype	0..1	ref	**Tags:** xml.sequenceOffset=40
targetMode	ModeDeclaration	0..1	ref	**Tags:** xml.sequenceOffset=60

Table F.3: PModeInSystemInstanceRef

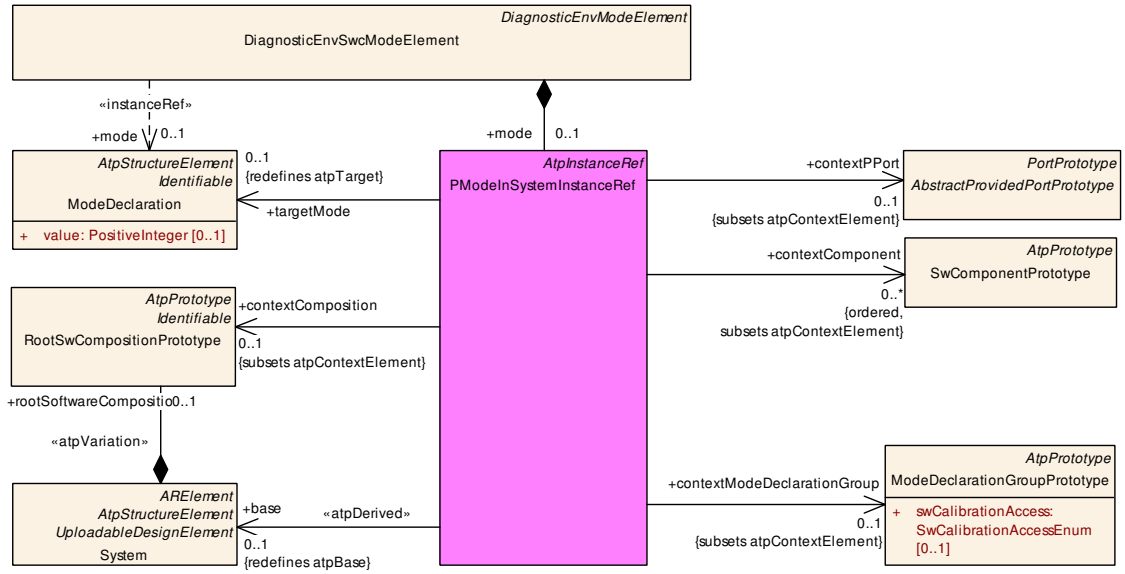

Figure F.4: Formal modeling of the comparison of a `ModeDeclarationGroupPrototype` with a `ModeDeclaration`

G Upstream Mapping

The content of this appendix chapter is *informative* in nature and shall **not** be considered as *normative* content.

G.1 Introduction

This chapter describes the mapping of the ECU Configuration parameters (M1 model) onto the meta-classes and attributes of the AUTOSAR upstream templates (System Template, SW Component Template, Diagnostic Extract, and ECU Resource Template).

The relationships between upstream templates and ECU Configuration are described in order to answer typical questions like:

- How shall a supplier use the information in a System Description in order to fulfill the needs defined by the systems engineer?
- How is a tool vendor supposed to generate an ECU Configuration Description out of ECU Extract of System Description?

Please note that the upstream mapping tables contain the following columns:

Column Name	Column Meaning
BSW Module	Name of BSW module
BSW Context	Reference to parameter container
BSW Parameter	Name of the BSW parameter
BSW Type	Type of parameter
BSW Description	Description from the configuration document
Template Description	Class note or attribute note of the M2 model element
M2 Parameter	Name of the upstream template model element
Mapping Rule	Textual description on how to transform between M2 and BSW domains
Mapping Type	One of: **local** no mapping needed since parameter local to BSW **partial** some data can be automatically mapped but not all **full** all data can be automatically mapped
Mapping Status	Indication of life-cycle status of the mapping
ECUC Parameter ID	ID of the parameter in the respective SWS document (may be empty if the mapping is owned by an enumeration literal)

Table G.1: Upstream mapping table columns

G.2 Dcm

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsd	
BSW Parameter		**BSW Type**
DcmDsdServiceTable		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container contains the configuration (DSD parameters) for a Service Identifier Table. Note: It is allowed to add OBD services to a DcmDsdServiceTable related to a UDS Protocol. But it is not allowed to add UDS services to a DcmDsdServiceTable related to an OBD Protocol.		
Template Description		
This meta-class represents a model of a diagnostic service table, i.e. the UDS services applicable for a given ECU.		
M2 Parameter		
DiagnosticServiceTable		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dcm_00732]

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService	
BSW Parameter		**BSW Type**
DcmDsdServiceRoleRef		ECUC-REFERENCE-DEF
BSW Description		
Reference to DcmDspAuthenticationRow that defines a role in that the service is allowed to be executed.		
Template Description		
This attribute allows for the specification of the position of the enclosing role in a bitfield of roles.		
M2 Parameter		
DiagnosticAuthRole.bitPosition		
Mapping Rule		**Mapping Type**
Applicable if the current access permission maps to a service where only the service ID shall get a assigned role. The value of DcmDsdServiceRole is calculated based on the bits of all referenced DiagnosticAuthRole.bitPosition, where a value of 2^bitPosition is used for each DiagnosticAuth Role.bitPosition. The accumulated value represents a bitfield that has a bit set for each Diagnostic AuthRole.bitPosition.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dcm_01139]

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService	
BSW Parameter		**BSW Type**
DcmDsdSidTabSecurityLevelRef		ECUC-REFERENCE-DEF
BSW Description		
Reference to a Security Level in which the service is allowed to be executed. Multiple references are allowed for a service. Please refer to ISO 14229-1, ISO 15031-5 and chapter "Verification of the Service Security Access levels." If there is no reference configured, no service security verification shall be performed.		
Template Description		

▽

△

This represents the associated DiagnosticSecurityLevels	
M2 Parameter	
DiagnosticAccessPermission.securityLevel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00733]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService
BSW Parameter	**BSW Type**
DcmDsdSidTabServiceId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Identifier of the service. The possible service identifiers are defined in ISO 14229-1 and ISO 15031-5.	
Template Description	
This meta-class provides the ability to define common properties that are shared among all instances of sub-classes of DiagnosticServiceInstance.	
M2 Parameter	
DiagnosticServiceClass	
Mapping Rule	**Mapping Type**
Service identifiers of the used DiagnosticServiceClass	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00735]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService
BSW Parameter	**BSW Type**
DcmDsdSidTabSessionLevelRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a Session Level in which the service is allowed to be executed. Multiple references are allowed for a service. Please refer to ISO 14229-1, ISO 15031-5 and chapter "Verification of the Diagnostic Session". If there is no reference configured, no diagnostic session verification shall be performed.	
Template Description	
This represents the associated DiagnosticSessions	
M2 Parameter	
DiagnosticAccessPermission.diagnosticSession	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00734]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService
BSW Parameter	**BSW Type**
DcmDsdSidTabSubfuncAvail	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Information about whether the service has subfunctions or not. This parameter is used for the handling of the "suppressPos RspMsgIndicationBit" as defined in ISO 14229-1, which can be used as a reference for the configuration. true - service has subfunctions, suppressPosRspMsgIndicationBit is available false - service has no subfunctions, suppressPosRspMsgIndicationBit is not available	
Template Description	
The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.	
M2 Parameter	
Identifiable.category	
Mapping Rule	**Mapping Type**
Stanadardized sub-functions of diagnostic services are mainly identified by the category. There are further specific attributes in the meta-model that allow for handling custom subfunctions,	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00737]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService/DcmDsdSubService
BSW Parameter	**BSW Type**
DcmDsdSubServiceFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Callback function of the ECU Supplier specific component for the particular service. The function's prototype is as described for <Module>_<DiagnosticService>_<SubService>. If this parameter is not configured, the subservice is handled Dcm-internally.	
Template Description	
This attribute shall be used to define a custom sub-function number if none of the standardized values of category shall be used.	
M2 Parameter	
DiagnosticComControl.customSubFunctionNumber	
Mapping Rule	**Mapping Type**
The existence of a custom subfunction number shall trigger the creation of a custom processor.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00942]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService/DcmDsdSubService
BSW Parameter	**BSW Type**
DcmDsdSubServiceRoleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to DcmDspAuthenticationRow that defines a role in that the service with this subfunction is allowed to be executed.	
Template Description	
This attribute allows for the specification of the position of the enclosing role in a bitfield of roles.	

△

M2 Parameter	
DiagnosticAuthRole.bitPosition	
Mapping Rule	**Mapping Type**
Applicable if the current access permission maps to a service where the subfunction of service ID and shall get a assigned role. The value of DcmDsdSubServiceRoleis is calculated based on the bits of all referenced DiagnosticAuthRole.bitPosition, where avalue of 2^bitPosition is used for each DiagnosticAuthRole.bitPosition. The accumulated value represents a bitfield that has a bit set for each DiagnosticAuthRole.bitPosition.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01140]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService/DcmDsdSubService
BSW Parameter	**BSW Type**
DcmDsdSubServiceSecurityLevelRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a Security Level in which the subservice is allowed to be executed. Multiple references are allowed for a subservice. Please refer to ISO 14229-1, ISO 15031-5 and chapter "Verification of the Service Security Access levels." If there is no reference configured, no subservice security verification shall be performed.	
Template Description	
This represents the associated DiagnosticSecurityLevels	
M2 Parameter	
DiagnosticAccessPermission.securityLevel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00812]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService/DcmDsdSubService
BSW Parameter	**BSW Type**
DcmDsdSubServiceSessionLevelRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a Session Level in which the subservice is allowed to be executed. Multiple references are allowed for a subservice. Please refer to ISO 14229-1, ISO 15031-5 and chapter "Verification of the Diagnostic Session". If there is no reference configured, no diagnostic session verification shall be performed.	
Template Description	
This represents the associated DiagnosticSessions	
M2 Parameter	
DiagnosticAccessPermission.diagnosticSession	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00804]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslDiagResp
BSW Parameter	**BSW Type**
DcmDslDiagRespMaxNumRespPend	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of negative responses with response code 0x78 (requestCorrectlyReceivedResponsePending) allowed for a request. If Dcm reaches this limit, an automatic 0x10 (generalReject) final response will be trasmitted and the service processing will be cancelled. Value 0x00 means that no NRC 0x78 response will be transmitted.	
Template Description	
Maximum number of negative responses with response code 0x78 (requestCorrectlyReceived-ResponsePending) allowed per request. DCM will send a negative response with response code 0x10 (generalReject), in case the limit value gets reached. Value 0xFF means that no limit number of NRC 0x78 response apply.	
M2 Parameter	
DiagnosticCommonProps.maxNumberOfRequestCorrectlyReceivedResponsePending	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00693]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslDiagResp
BSW Parameter	**BSW Type**
DcmDslDiagRespOnSecondDeclinedRequest	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines the reaction upon a second request (ClientB) that can not be processed (e.g. due to priority assessment). TRUE: when the second request (Client B) can not be processed, it shall be answered with NRC21 BusyRepeatRequest. FALSE: when the second request (Client B) can not be processed, it shall not be responded.	
Template Description	
Defines the reaction upon a second request (ClientB) that can not be processed (e.g. due to priority assessment). TRUE: when the second request (Client B) can not be processed, it shall be answered with NRC21 BusyRepeatRequest. FALSE: when the second request (Client B) can not be processed, it shall not be responded.	
M2 Parameter	
DiagnosticCommonProps.responseOnSecondDeclinedRequest	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00914]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol
BSW Parameter	**BSW Type**
DcmDslProtocolRow	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration of one particular diagnostic protocol used in Dcm.	
Template Description	
This meta-class represents the ability to define a diagnostic protocol.	

▽

△

M2 Parameter	
DiagnosticProtocol	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00695]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection
BSW Parameter	**BSW Type**
DcmDslProtocolComMChannelRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the ComMChannel on which the DcmDslProtocolRxPdu is received and the DcmDslProtocolTxPdu is transmitted.	
Template Description	
The CommunicationCluster is the main element to describe the topological connection of communicating ECUs. A cluster describes the ensemble of ECUs, which are linked by a communication medium of arbitrary topology (bus, star, ring, ...). The nodes within the cluster share the same communication protocol, which may be event-triggered, time-triggered or a combination of both. A CommunicationCluster aggregates one or more physical channels.	
M2 Parameter	
CommunicationCluster	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00952]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection/DcmDslProtocolRx/DcmDslProtocolRxAddrType
BSW Parameter	**BSW Type**
DCM_FUNCTIONAL_TYPE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
FUNCTIONAL = 1 to n communication	
Template Description	
Reference to functional request messages.	
M2 Parameter	
DiagnosticConnection.functionalRequest	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection/DcmDslProtocolRx/DcmDslProtocolRxAddrType

BSW Parameter	BSW Type
DCM_PHYSICAL_TYPE	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
PHYSICAL = 1 to 1 communications using physical addressing

Template Description
Reference to a physical request message.

M2 Parameter
DiagnosticConnection.physicalRequest

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection/DcmDslProtocolRx

BSW Parameter	BSW Type
DcmDslProtocolRxPduRef	ECUC-REFERENCE-DEF

BSW Description
Reference to a Pdu in EcuC that is used for this reception channel.

Template Description
The IPdu (Interaction Layer Protocol Data Unit) element is used to sum up all Pdus that are routed by the PduR.

M2 Parameter
IPdu

Mapping Rule	Mapping Type
Reference to IPdu of xxxTpConnection for DiagnosticConnection.physicalRequest / DiagnosticConnection.functionalRequest	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00770]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection

BSW Parameter	BSW Type
DcmDslProtocolTx	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description
This container contains the configuration parameters of a transmission channel in a diagnostic connection. The PDU referenced by this transmission channel can produce meta data items of type TARGET_ADDRESS_16 and SOURCE_ADDRESS_16.

Template Description
In the vast majority of cases a response is required. However, there are also cases where providing the response is not possible and/or not allowed.

M2 Parameter
DiagnosticConnection.response

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00711]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection/DcmDslProtocolTx
BSW Parameter	**BSW Type**
DcmDslProtocolTxPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a Pdu in EcuC that is used for this transmission channel.	
Template Description	
The IPdu (Interaction Layer Protocol Data Unit) element is used to sum up all Pdus that are routed by the PduR.	
M2 Parameter	
IPdu	
Mapping Rule	**Mapping Type**
Reference to IPdu of xxxTpConnection for DiagnosticConnection.response	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00772]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslPeriodicTransmission/DcmDslPeriodicConnection
BSW Parameter	**BSW Type**
DcmDslPeriodicTxPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a Pdu in EcuC that is used for this periodic transmission channel.	
Template Description	
The IPdu (Interaction Layer Protocol Data Unit) element is used to sum up all Pdus that are routed by the PduR.	
M2 Parameter	
IPdu	
Mapping Rule	**Mapping Type**
Reference to IPdu of xxxTpConnection in case of DiagnosticConnection.periodicResponseTp or IPdu of PduTriggering in case of DiagnosticConnection.periodicResponseUudt	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00742]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow
BSW Parameter	**BSW Type**
DcmDslProtocolPriority	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

Protocol priority used during protocol preemption. A higher priority protocol may preempt a lower priority protocol. Lower numeric values represent higher protocol priority: 0 - Highest protocol priority 255 - Lowest protocol priority
Template Description
This represents the priority of the diagnostic protocol in comparison to other diagnostic protocols. Lower numeric values represent higher protocol priority: • 0 - Highest protocol priority • 255 - Lowest protocol priority
M2 Parameter
DiagnosticProtocol.priority

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00699]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolType

BSW Parameter	**BSW Type**
DCM_PERIODICTRANS_ON_CAN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Periodic Transmission on CAN	
Template Description	
AbstractCanPhysicalChannel: Abstract class that is used to collect the common CAN PhysicalChannel attributes. **DiagnosticConnection.periodicResponseUudt:** Reference to UUDT responses.	
M2 Parameter	
AbstractCanPhysicalChannel, DiagnosticConnection.periodicResponseUudt	

Mapping Rule	**Mapping Type**
If DiagnosticConnection.periodicResponseUudt exists and PhysicalChannel given as AbstractCan PhysicalChannel.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolType

BSW Parameter	**BSW Type**
DCM_PERIODICTRANS_ON_IP	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Periodic Transmission on Internet Protocol	
Template Description	

▽

△

StaticSocketConnection: Definition of static SocketConnection between the Socket that is defined by the aggregating SocketAddress and the remote Address. **DiagnosticConnection.periodicResponseUudt:** Reference to UUDT responses. **EthernetPhysicalChannel:** The EthernetPhysicalChannel represents a VLAN or an untagged channel. An untagged channel is modeled as an Ethernet PhysicalChannel without an aggregated VLAN.
M2 Parameter
StaticSocketConnection, DiagnosticConnection.periodicResponseUudt, EthernetPhysicalChannel

Mapping Rule	**Mapping Type**
If DiagnosticConnection.periodicResponseUudt exists and PhysicalChannel given as Ethernet PhysicalChannel.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolType

BSW Parameter	**BSW Type**
DCM_UDS_ON_CAN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
UDS on CAN (ISO15765-3; ISO14229-1)	
Template Description	
DiagnosticConnection.physicalRequest: Reference to a physical request message. **CanTpConnection:** A connection identifies the sender and the receiver of this particular communication. The CanTp module routes a Pdu through this connection. atpVariation: Derived, because TpNode can vary.	
M2 Parameter	
DiagnosticConnection.physicalRequest, CanTpConnection	

Mapping Rule	**Mapping Type**
In case DiagnosticConnection.physicalRequest exists and TpConnectionIdent.ident belongs to CanTpConnection	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolType

BSW Parameter	**BSW Type**
DCM_UDS_ON_FLEXRAY	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
UDS on FlexRay (Manufacturer specific; ISO14229-1)	
Template Description	

▽

△

DiagnosticConnection.physicalRequest: Reference to a physical request message. **FlexrayTpConnection:** A connection identifies the sender and the receiver of this particular communication. The FlexRayTp module routes a Pdu through this connection. In a System Description the references to the PduPools are mandatory. In an ECU Extract these references can be optional: On unicast connections these references are always mandatory. On multicast the txPduPool is mandatory on the sender side. The rxPduPool is mandatory on the receiver side. On Gateway ECUs both references are mandatory.	
M2 Parameter	
DiagnosticConnection.physicalRequest, FlexrayTpConnection	
Mapping Rule	**Mapping Type**
In case DiagnosticConnection.physicalRequest exists and TpConnectionIdent.ident belongs to FlexRayTpConnection	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolType
BSW Parameter	**BSW Type**
DCM_UDS_ON_IP	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
UDS on Internet Protocol (Manufacturer specific; ISO14229-1)	
Template Description	
DiagnosticConnection.physicalRequest: Reference to a physical request message. **StaticSocketConnection:** Definition of static SocketConnection between the Socket that is defined by the aggregating SocketAddress and the remote Address.	
M2 Parameter	
DiagnosticConnection.physicalRequest, StaticSocketConnection	
Mapping Rule	**Mapping Type**
In case DiagnosticConnection.physicalRequest exists and TpConnectionIdent.ident belongs to a SocketConnection	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspAuthentication
BSW Parameter	**BSW Type**
DcmDspAuthenticationDeauthenticatedRoleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to DcmDspAuthenticationRow that defines a role in that is used as deauthenticated role.	
Template Description	
This attribute allows for the specification of the position of the enclosing role in a bitfield of roles.	
M2 Parameter	
DiagnosticAuthRole.bitPosition	
Mapping Rule	**Mapping Type**

▽

△

The value of each DcmDspAuthenticationDeauthenticatedRole is calculated based on the bits of all referenced DiagnosticAuthRole.bitPosition with DiagnosticAuthRole.isDefault = true, where a value of 2^bitPosition is used for each DiagnosticAuthRole.bitPosition. The accumulated value represents a bitfield that has a bit set for each DiagnosticAuthRole.bitPosition.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01153]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspAuthentication
BSW Parameter	**BSW Type**
DcmDspAuthenticationDefaultSessionTimeOut	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the number of seconds after which the Dcm makes a transition to deauthenticated state, in case of no active communication.	
Template Description	
This attribute defines the time (in seconds) that the authentication state is maintained in default-session if there is no communication from the authenticated client.	
M2 Parameter	
DiagnosticCommonProps.authenticationTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01161]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspAuthentication/DcmDspAuthenticationTransmitCertificate
BSW Parameter	**BSW Type**
DcmDspAuthenticationTransmitCertificateEvaluationId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Certificate evaluation id.	
Template Description	
This attributes represents the ID of the certificate evaluation.	
M2 Parameter	
DiagnosticAuthTransmitCertificateEvaluation.evaluationId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01242]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspAuthentication/DcmDspAuthenticationTransmitCertificate
BSW Parameter	**BSW Type**
DcmDspAuthenticationTransmitCertificateFunction	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Provides a fixed semantics to this certificate that will lead to predefined actions in the Dcm.	

▽

△

Template Description	
This meta-class represents the ability to configure a certificate evaluation in the context of a diagnostic authentication.	
M2 Parameter	
DiagnosticAuthTransmitCertificateEvaluation	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01244]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspAuthentication/DcmDspAuthenticationTransmitCertificate
BSW Parameter	**BSW Type**
DcmDspAuthenticationTransmitCertificateRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a KeyMCertificate used to handle the certificate.	
Template Description	
This reference identifies the description of the applicable crypto certificate.	
M2 Parameter	
DiagnosticAuthTransmitCertificateMapping.cryptoServiceCertificate	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01243]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspComControl
BSW Parameter	**BSW Type**
DcmDspComControlAllChannel	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Collection of ComM channels which shall be controlled if all networks are addressed.	
Template Description	
DiagnosticComControlClass.allChannels: This reference represents the semantics that all available channels shall be affected. It is still necessary to refer to individual CommunicatuionClusters because there could be private CommunicationClusters in the System Extract that are not subject to the service "communication control". By referring to the applicable CommunicationClusters it can be made sure that only the affected CommunicationClusters are accessed. **DiagnosticComControlClass.allPhysicalChannels:** This reference represents the semantics that all available channels shall be affected. It is still necessary to refer to individual EthernetPhysicalChannels because there could be private VLANs (and thus private EthernetPhysicalChannels) in the System Extract that are not subject to the service "communication control". By referring to the applicable EthernetPhysicalChannels it can be made sure that only the affected EthernetPhysicalChannels are accessed.	
M2 Parameter	
DiagnosticComControlClass.allChannels, DiagnosticComControlClass.allPhysicalChannels	
Mapping Rule	**Mapping Type**

▽

△

For all bus systems, except Ethernet: Create one DcmDspComControlAllChannel for each DiagnosticComControl.allChannels. Add reference DcmDspAllComControlChannelRef and let it refer to the ComMChannel that represents the respective CommunicationCluster that is referenced in the role DiagnosticComControlClass.all Channels. For Ethernet Bus Systems: Create one DcmDspComControlAllChannel for each element of DiagnosticComControlClass.all PhysicalChannels. Add reference DcmDspAllComControlChannelRef and let it refer to the Com MChannel that represents the respective PhysicalChannel that is referenced in the role Diagnostic ComControlClass.allPhysicalChannels.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00901]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspComControl
BSW Parameter	**BSW Type**
DcmDspComControlSpecificChannel	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Assigns subnet number to ComM channel which will be controlled.	
Template Description	
DiagnosticComControlClass.specificChannel: This represents the ability to add additional attributes to the case that only specific channels are supposed to be considered, **DiagnosticComControlSpecificChannel.specificPhysicalChannel:** This represents the affected specific EthernetPhysicalChannel.	
M2 Parameter	
DiagnosticComControlClass.specificChannel, DiagnosticComControlSpecificChannel.specificPhysicalChannel	

Mapping Rule	**Mapping Type**
For all bus systems, except Ethernet: Create one DcmDspComControlSpecificChannel for each DiagnosticComControlSpecific Channel.specificChannel. Add reference DcmDspSpecificComControlChannelRef and let it refer to the ComMChannel that represents the respective CommunicationCluster that is referenced in the role DiagnosticComControlClass.specificChannel. For Ethernet Bus Systems: Create one DcmDspComControlSpecificChannel for each DiagnosticComControlSpecific Channel.specificPhysicalChannel. Add reference DcmDspSpecificComControlChannelRef and let it refer to the ComMChannel that represents the respective PhysicalChannel that is referenced in the role DiagnosticComControlSpecificChannel.specificPhysicalChannel.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00903]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspComControl/DcmDspComControlSpecificChannel
BSW Parameter	**BSW Type**
DcmDspSubnetNumber	ECUC-INTEGER-PARAM-DEF
BSW Description	
Subnet Number which controls the specific ComMChannel.	
Template Description	

▽

△

This represents the applicable subnet number (which is an arbitrary number ranging from 1..14)	
M2 Parameter	
DiagnosticComControlSpecificChannel.subnetNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00905]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspComControl
BSW Parameter	**BSW Type**
DcmDspComControlSubNode	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container gives information about the node identification number and the ComM channel used to address a sub-network.	
Template Description	
DiagnosticComControlSubNodeChannel.subNodeChannel: This represents the affected CommunicationCluster in the role subNodeChannel **DiagnosticComControlSubNodeChannel.subNodePhysicalChannel:** This represents the affected sub-node EthernetPhysicalChannel.	
M2 Parameter	
DiagnosticComControlSubNodeChannel.subNodeChannel, DiagnosticComControlSubNodeChannel.subNodePhysicalChannel	
Mapping Rule	**Mapping Type**
For all bus systems, except Ethernet: Create one DcmDspComControlSubNodeChannel for each DiagnosticComControlSubNode Channel.subNodeChannel. Add reference DcmDspComControlSubNodeComMChannelRef and let it refer to the ComMChannel that represents the respective CommunicationCluster that is referenced in the role DiagnosticComControlClass.subNodeChannel. For Ethernet Bus Systems: Create one DcmDspComControlSubNodeChannel for each DiagnosticComControlSubNode Channel.subnodePhysicalChannel. Add reference DcmDspComControlSubNodeComMChannel Ref and let it refer to the ComMChannel that represents the respective PhysicalChannel that is referenced in the role DiagnosticComControlSubNodeChannel.subNodePhysicalChannel.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01033]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspComControl/DcmDspComControlSubNode
BSW Parameter	**BSW Type**
DcmDspComControlSubNodeId	ECUC-INTEGER-PARAM-DEF
BSW Description	
The node identification number DcmDspComControlSubNodeId is addressed by the CommunicationControl (0x28) request.	
Template Description	
This represents the applicable subNode number. The value corresponds to the request message parameter node IdentificationNumber of diagnostic service CommunicationControl (0x28).	
M2 Parameter	
DiagnosticComControlSubNodeChannel.subNodeNumber	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01031]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp
BSW Parameter	**BSW Type**
DcmDspCommonAuthorization	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration (parameters) for the common Authorization being equal for several services / sub-services.	
Template Description	
This represents an instance of the "Routine Control" diagnostic service.	
M2 Parameter	
DiagnosticRoutineControl	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01025]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp
BSW Parameter	**BSW Type**
DcmDspDDDIDcheckPerSourceDID	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines the check for session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22). true: Dcm module shall check the session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF false: Dcm module shall not check the session, security and mode dependencies per source DIDs with a ReadDataBy Identifier (0x22) with DID in the range 0xF200 to 0xF3FF	
Template Description	
If set to TRUE, the Dcm module shall check the session, security and mode dependencies per source DIDs with a ReadData ByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF. If set to FALSE. the Dcm module shall not check the session, security and mode dependencies per source DIDs with a Read DataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF.	
M2 Parameter	
DiagnosticDynamicallyDefineDataIdentifierClass.checkPerSourceId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00966]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspDataByteSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the array length in bytes or the maximum array length for variable datalengths.	
Template Description	
DiagnosticDataElement.maxNumberOfElements: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.	
M2 Parameter	
DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
S/R via array: DcmDspDataByteSize= maxNumberOfElements * (baseTypeSize / 8) C/S of FNC callback: DcmDspDataByteSize= maxNumberOfElements Note: 8 is the baseType Size of UINT8	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01106]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspDataConditionCheckReadFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Function name to demand application if the conditions (e.g. System state) to read the DID are correct. (ConditionCheck Read-function). Multiplicity shall be equal to parameter DcmDspDataReadFnc. This parameter is related to the interface Xxx_ConditionCheck Read.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00677]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspDataEndianness	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the endianness of the data belonging to a DID in a diagnostic request or response message.	
Template Description	



△

This attribute specifies the byte order of the base type.	
M2 Parameter	
BaseTypeDirectDefinition.byteOrder	
Mapping Rule	**Mapping Type**
baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the Diagnostic Parameter with the role DiagnosticDataIdentifier.dataElement	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00986]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspDataFreezeCurrentStateFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Function name to request to application to freeze the current state of an IOControl. (FreezeCurrentState-function). This parameter is related to the interface Xxx_FreezeCurrentState.	
Template Description	
DiagnosticIoControlNeeds.freezeCurrentStateSupported: This attribute determines, if the referenced port supports temporary freezing of I/O value. **DiagnosticServiceSwMapping.mappedBswServiceDependency:** This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticIoControlNeeds.freezeCurrentStateSupported, DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00674]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspDataReadDataLengthFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Function name to request from application the data length of a DID. (ReadDataLength-function). This parameter is related to the interface Xxx_ReadDataLength.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00671]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspDataReadFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Function name to request from application the data value of a DID. (ReadData-function). This parameter is related to the interface Xxx_ReadData.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00669]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspDataResetToDefaultFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Function name to request to application to reset an IOControl to default value. (ResetToDefault-function). This parameter is related to the interface Xxx_ResetToDefault.	
Template Description	
DiagnosticIoControlNeeds.resetToDefaultSupported: This represents a flag for the existence of the ResetToDefault operation in the service interface. **DiagnosticServiceSwMapping.mappedBswServiceDependency:** This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticIoControlNeeds.resetToDefaultSupported, DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00673]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspDataReturnControlToEcuFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Function name to request to application to return control to ECU of an IOControl. (ReturnControlToECU-function). This parameter is related to the interface Xxx_ReturnControlToECU.	
Template Description	

▽

△

This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00672]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspDataShortTermAdjustmentFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Function name to request to application to adjust the IO signal. (ShortTermAdjustment-function). This parameter is related to the interface Xxx_ShortTermAdjustment.	
Template Description	
DiagnosticIoControlNeeds.shortTermAdjustmentSupported: This attribute determines, if the referenced port supports temporarily setting of I/O value to a specific value provided by the diagnostic tester. **DiagnosticServiceSwMapping.mappedBswServiceDependency:** This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticIoControlNeeds.shortTermAdjustmentSupported, DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00675]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspDataType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Provide the implementation data type of data belonging to a DID.	
Template Description	

▽

△

DiagnosticEnvDataElementCondition:
This meta-class represents the ability to formulate a diagnostic environment condition based on the value of a data element owned by the application software.

SwDataDefProps:
This class is a collection of properties relevant for data objects under various aspects. One could consider this class as a "pattern of inheritance by aggregation". The properties can be applied to all objects of all classes in which SwDataDefProps is aggregated.

Note that not all of the attributes or associated elements are useful all of the time. Hence, the process definition (e.g. expressed with an OCL or a Document Control Instance MSR-DCI) has the task of implementing limitations.

SwDataDefProps covers various aspects:

- Structure of the data element for calibration use cases: is it a single value, a curve, or a map, but also the recordLayouts which specify how such elements are mapped/converted to the DataTypes in the programming language (or in AUTOSAR). This is mainly expressed by properties like swRecordLayout and swCalprmAxisSet
- Implementation aspects, mainly expressed by swImplPolicy, swAddrMethod, baseType, implementationDataType and additionalNativeTypeQualifier
- Access policy for the MCD system, mainly expressed by swCalibrationAccess
- Semantics of the data element, mainly expressed by compuMethod and/or unit, dataConstr, invalidValue
- Code generation policy provided by swRecordLayout

SwBaseType:
This meta-class represents a base type used within ECU software.

M2 Parameter	
DiagnosticEnvDataElementCondition, SwDataDefProps, SwBaseType	
Mapping Rule	**Mapping Type**
Maps to DiagnosticEnvDataElementCondition.swDataDefProps.baseType: the details are explained in the upstream mappings of the individual enumeration literals.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00985]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
BOOLEAN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is boolean.	
Template Description	

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = BOOLEAN • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
DOUBLE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is double (64 bit float).	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
DOUBLE_N	ECUC-ENUMERATION-LITERAL-DEF

▽

△

BSW Description
Type of the data is double (64 bit float) array.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
FLOAT	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is float.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
FLOAT_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is float array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**

▽

△

SINT16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeSize, BaseTypeDirectDefinition.baseTypeEncoding, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
SINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeSize, BaseTypeDirectDefinition.baseTypeEncoding, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001)* arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
SINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeSize, BaseTypeDirectDefinition.baseTypeEncoding, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = 2C • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType

BSW Parameter	BSW Type
SINT32_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint32 array.

Template Description

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeSize`, `BaseTypeDirectDefinition.baseTypeEncoding`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = 2C • `baseTypeSize` = 32 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01001) • `arraySizeSemantics` either does not exist or exists and is set to `fixedSize` (cf. TPS_DEXT_01001) Derivation from `fixedLength` =1 possible.	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType

BSW Parameter	BSW Type
SINT64	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint64.

Template Description

▽

△

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = NONE • `baseTypeSize` = 64 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist Derivation from `fixedLength` =1 possible.	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType

BSW Parameter	BSW Type
SINT64_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint64 array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`

Mapping Rule	Mapping Type

▽

△

• baseTypeEncoding = 2C • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
SINT8	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeSize, BaseTypeDirectDefinition.baseTypeEncoding, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = 2C • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
SINT8_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint8 array.	

▽

△

Template Description
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed.
M2 Parameter
BaseTypeDirectDefinition.baseTypeSize, BaseTypeDirectDefinition.baseTypeEncoding, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
• baseTypeEncoding = 2C • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001)* arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType

BSW Parameter	BSW Type
UINT16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
UINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE,UTF-16 • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
UINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`	
Mapping Rule	**Mapping Type**
• `baseTypeEncoding` = NONE • `baseTypeSize` = 32 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist Derivation from `fixedLength` =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
UINT32_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint32 array.	
Template Description	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
• baseTypeEncoding = NONE,UTF-32 • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
UINT64	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is uint64.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics

▽

△

Mapping Rule	Mapping Type
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
UINT64_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint64 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
UINT8	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Type of the data is uint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
UINT8_DYN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint8 array with dynamic length.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE,WINDOWS-1252,UTF-8 • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to variableSize (cf. TPS_DEXT_01001) Derivation from fixedLength =0 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
UINT8_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspDataUsePort	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines which interface shall be used to access the data.	
Template Description	
This attribute controls whether interaction requires the software-component to react synchronously on a request or whether it processes the request in background but still the DCM has to issue the call again to eventually obtain the result of the request.	
M2 Parameter	
DiagnosticValueNeeds.processingStyle	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00713]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort
BSW Parameter	**BSW Type**
USE_DATA_ASYNCH_CLIENT_SERVER	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The DCM will access the Data using an R-Port requiring a asynchronous ClientServerInterface DataServices_{Data}. The R-Port is named DataServices_{Data} where {Data} is the name of the container DcmDspData.	
Template Description	
The software-component processes the request in background but still the Dcm has to issue the call again to eventually obtain the result of the request.	
M2 Parameter	
DiagnosticProcessingStyleEnum.processingStyleAsynchronous	
Mapping Rule	**Mapping Type**
DiagnosticServiceSwMapping is having a SwcServiceDependency and ServiceNeeds::Diagnostic ProcessingStyleEnum is equal to processingStyleAsynchronous	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort
BSW Parameter	**BSW Type**
USE_DATA_ASYNCH_CLIENT_SERVER_ERROR	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The Dcm will access the Data using an R-Port requiring a asynchronous ClientServerInterface DataServices_{Data}. The parameter ErrorCode can be returned to allow the application to trigger a negative response during the operation. The R-Port is named DataServices_{Data} where {Data} is the name of the container DcmDspData.	
Template Description	
The software-component processes the request in background but still the Dcm has to issue the call again to eventually obtain the result of the request or handle error code.	
M2 Parameter	
DiagnosticProcessingStyleEnum.processingStyleAsynchronousWithError	



△

Mapping Rule	Mapping Type
DiagnosticServiceSwMapping is having a SwcServiceDependency and ServiceNeeds::DiagnosticProcessingStyleEnum is equal to processingStyleAsynchronousWithError	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort
BSW Parameter	**BSW Type**
USE_DATA_ASYNCH_FNC	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The DCM will access the Data using the functions that are defined in parameters of type EcucFunctionNameDef (but without DcmDspDataReadDataLengthFnc) in the DcmDspData container. DCM_E_PENDING return is allowed. OpStatus is existing as IN parameter.	
Template Description	
The software-component processes the request in background but still the Dcm has to issue the call again to eventually obtain the result of the request.	
M2 Parameter	
DiagnosticProcessingStyleEnum.processingStyleAsynchronous	
Mapping Rule	**Mapping Type**
DiagnosticServiceSwMapping is having a BswServiceDependency and ServiceNeeds::DiagnosticProcessingStyleEnum is equal to processingStyleAsynchronous	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort
BSW Parameter	**BSW Type**
USE_DATA_SENDER_RECEIVER	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The DCM will access the Data using an Port requiring a SenderReceiverInterface (with isService=false) Data Services_{Data}. The Port is named DataServices_{Data} where {Data} is the name of the container DcmDspData.	
Template Description	
This represents the ability to define a mapping of a diagnostic service to a software-component. This kind of service mapping is applicable for the usage of SenderReceiverInterfaces or event/notifier semantics in Service Interfaces on the adaptive platform.	
M2 Parameter	
DiagnosticServiceDataMapping	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort
BSW Parameter	**BSW Type**

▽

△

USE_DATA_SENDER_RECEIVER_AS_SERVICE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The DCM will access the Data using an service Port requiring a SenderReceiverInterface (with isService=true) Data Services_{Data} . The Port is namedDataServices_{Data} where {Data} is the name of the container DcmDspData.	
Template Description	
This represents the ability to define a mapping of a diagnostic service to a software-component. This kind of service mapping is applicable for the usage of SenderReceiverInterfaces or event/notifier semantics in Service Interfaces on the adaptive platform.	
M2 Parameter	
DiagnosticServiceDataMapping	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspDataWriteFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Function name to request application to write the data value of a DID. (WriteData-function). This parameter is related to the interface Xxx_WriteData.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00670]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDiagnosisScaling/DcmDspAlternativeDataType
BSW Parameter	**BSW Type**
DcmDspTextTableMapping	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.	
Template Description	

▽

△

This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.	
M2 Parameter	
CompuMethod	
Mapping Rule	**Mapping Type**
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00999]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDiagnosisScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspDiagnosisRepresentationDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The data value in the diagnosis representation.	
Template Description	
This represents a textual constant in the computation method.	
M2 Parameter	
CompuConstTextContent.vt	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01001]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDiagnosisScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspInternalDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The ECU internal data value.	
Template Description	
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit:** This specifies the upper limit of a of the scale.	
M2 Parameter	
CompuScale.lowerLimit, CompuScale.upperLimit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01000]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspDidDataSupportInfo	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container defines the supported information.	
Template Description	
DiagnosticParameter.supportInfo: This attribute represents the ability to define which bit of the support info byte is representing this part of the PID. **DiagnosticDataIdentifier.supportInfoByte:** This attribute represents the supported information associated with the DiagnosticDataIdentifier.	
M2 Parameter	
DiagnosticParameter.supportInfo, DiagnosticDataIdentifier.supportInfoByte	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01104]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspExternalSRDataElementClass/DcmDataElementInstance
BSW Parameter	**BSW Type**
DcmDataElementInstanceRef	ECUC-INSTANCE-REFERENCE-DEF
BSW Description	
Instance Reference to the primitive or array data which shall be read or written. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationPrimitiveDataType of category VALUE or BOOLEAN or ApplicationArrayDataType or if the AutosarDataPrototype is typed with a ImplementationDataType of category VALUE, ARRAY or TYPE_REFERENCE that in turn boils down to VALUE or ARRAY	
Template Description	
DiagnosticServiceDataMapping.mappedDataElement: This represents the dataElement in the application software that is accessed for diagnostic purpose. This role is applicable on the classic platform. **DiagnosticEnvDataElementCondition:** This meta-class represents the ability to formulate a diagnostic environment condition based on the value of a data element owned by the application software.	
M2 Parameter	
DiagnosticServiceDataMapping.mappedDataElement, DiagnosticEnvDataElementCondition	
Mapping Rule	**Mapping Type**
DiagnosticServiceDataMapping maps to a primitive data.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00991]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspExternalSRDataElementClass/DcmSubElementInDataElementInstance
BSW Parameter	**BSW Type**
DcmSubElementInDataElementInstanceRef	ECUC-INSTANCE-REFERENCE-DEF
BSW Description	

▽

△

Instance Reference to the primitve sub-element (at any level) of composite data in a port which shall be read. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in Parameter Interfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationCompositeData Type.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
`DiagnosticServiceSwMapping.mappedBswServiceDependency`	
Mapping Rule	**Mapping Type**
DiagnosticServiceDataMapping maps to a primitive element within a composite data, where the AutosarDataPrototype is typed with a ApplicationCompositeDataType.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00990]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspOdxDataDescription	ECUC-ADD-INFO-PARAM-DEF
BSW Description	
Defines additional description for ODX documentation	
Template Description	
This specifies the long name of the object. Long name is targeted to human readers and acts like a headline.	
M2 Parameter	
`MultilanguageReferrable.longName`	
Mapping Rule	**Mapping Type**
Textual description that characterizes the DID element with respect to the ODX long name can be provided by means of the attribute long-Name.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00988]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp
BSW Parameter	**BSW Type**
DcmDspDataDefaultEndianness	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the default endianness belonging to a DID, RID or PID if the corresponding data does not define an endianness.	
Template Description	
Defines the default endianness of the data belonging to a DID or RID which is applicable if the DiagnosticDataElement does not define the endianness via the swDataDefProps.baseType attribute.	
M2 Parameter	
`DiagnosticCommonProps.defaultEndianness`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00987]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp
BSW Parameter	**BSW Type**
DcmDspDid	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration (parameters) of the DID.	
Template Description	
DiagnosticDataIdentifier: This meta-class represents the ability to model a diagnostic data identifier (DID) that is fully specified regarding the payload at configuration-time. **DiagnosticDynamicDataIdentifier:** This meta-class represents the ability to define a diagnostic data identifier (DID) at run-time.	
M2 Parameter	
DiagnosticDataIdentifier, DiagnosticDynamicDataIdentifier	
Mapping Rule	**Mapping Type**
If the DiagnosticDataIdentifier is referenced by DiagnosticDataByIdentifier, DiagnosticIOControl, or DiagnosticReadDataByPeriodicID,DiagnosticDataChangeTrigger	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00601]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid
BSW Parameter	**BSW Type**
DcmDspDidIdentifier	ECUC-INTEGER-PARAM-DEF
BSW Description	
2 byte Identifier of the DID Within each DcmConfigSet all DcmDspDidIdentifier values shall be unique.	
Template Description	
This is the numerical identifier used to identify the DiagnosticAbstractDataIdentifier in the scope of diagnostic workflow	
M2 Parameter	
DiagnosticAbstractDataIdentifier.id	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00602]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidSignal
BSW Parameter	**BSW Type**
DcmDspDataByteSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the array length in bytes or the maximum array length for variable datalengths.	
Template Description	

▽

△

DiagnosticDataElement.maxNumberOfElements: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.
M2 Parameter
DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type
S/R via array: DcmDspDataByteSize= maxNumberOfElements * (baseTypeSize / 8) C/S of FNC callback: DcmDspDataByteSize= maxNumberOfElements Note: 8 is the baseType Size of UINT8	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01106]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidSignal

BSW Parameter	BSW Type
DcmDspDidByteOffset	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the absolute byte offset of the data defined by DcmDspDidDataRef reference to DcmDspData container in the DID.	
Template Description	
This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticDataIdentifier, DiagnosticParameterIdentifier, or DiagnosticRoutineSubfunction.	
M2 Parameter	
DiagnosticParameter.bitOffset	

Mapping Rule	Mapping Type
bitOffset / 8	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01105]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidSignal

BSW Parameter	BSW Type
DcmDspDidSignalCompositeRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the contained signal.	
Template Description	
DiagnosticParameterIdent.subElement: This collection represents the subElements on the top level. **DiagnosticParameterElement.subElement:** This collection represents the sub-elements on the next lower level.	
M2 Parameter	
DiagnosticParameterIdent.subElement, DiagnosticParameterElement.subElement	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01232]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid
BSW Parameter	**BSW Type**
DcmDspDidSignalCompositePool	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Defines a collection of all contained signals for this root signal.	
Template Description	
This meta-class represents an element of a DiagnosticParameter if the DiagnosticParameter represents a structure.	
M2 Parameter	
DiagnosticParameterElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01231]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidSignalCompositePool/DcmDspDidSignal
BSW Parameter	**BSW Type**
DcmDspDataByteSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the array length in bytes or the maximum array length for variable datalengths.	
Template Description	
DiagnosticDataElement.maxNumberOfElements: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.	
M2 Parameter	
DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
S/R via array: DcmDspDataByteSize= maxNumberOfElements * (baseTypeSize / 8) C/S of FNC callback: DcmDspDataByteSize= maxNumberOfElements Note: 8 is the baseType Size of UINT8	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01106]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidSignalCompositePool/DcmDspDidSignal
BSW Parameter	**BSW Type**
DcmDspDidByteOffset	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the absolute byte offset of the data defined by DcmDspDidDataRef reference to DcmDspData container in the DID.	
Template Description	
This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticDataIdentifier, DiagnosticParameterIdentifier, or DiagnosticRoutineSubfunction.	
M2 Parameter	

△

DiagnosticParameter.bitOffset	
Mapping Rule	**Mapping Type**
bitOffset / 8	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01105]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidSignalCompositePool/DcmDspDidSignal
BSW Parameter	**BSW Type**
DcmDspDidSignalCompositeRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the contained signal.	
Template Description	
DiagnosticParameterIdent.subElement: This collection represents the subElements on the top level. **DiagnosticParameterElement.subElement:** This collection represents the sub-elements on the next lower level.	
M2 Parameter	
DiagnosticParameterIdent.subElement, DiagnosticParameterElement.subElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01232]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid
BSW Parameter	**BSW Type**
DcmDspDidSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Length of a DID in byte(s).	
Template Description	
This attribute indicates the size in bytes of the DiagnosticDataIdentifier.	
M2 Parameter	
DiagnosticDataIdentifier.didSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01099]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid
BSW Parameter	**BSW Type**
DcmDspDidSupportInfo	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

This container defines the support information to declare the usability of the data bytes within the DIDs	
Template Description	
This attribute represents the supported information associated with the DiagnosticDataIdentifier.	
M2 Parameter	
DiagnosticDataIdentifier.supportInfoByte	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01102]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo
BSW Parameter	**BSW Type**
DcmDspDDDIDMaxElements	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of source elements of a DDDID.	
Template Description	
This represents the maximum number of source elements of the dynamically created DID.	
M2 Parameter	
DiagnosticDynamicallyDefineDataIdentifier.maxSourceElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00970]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl
BSW Parameter	**BSW Type**
DcmDspDidControlRoleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to DcmDspAuthenticationRow that defines a role in that this IO can be be controlled.	
Template Description	
This attribute allows for the specification of the position of the enclosing role in a bitfield of roles.	
M2 Parameter	
DiagnosticAuthRole.bitPosition	
Mapping Rule	**Mapping Type**
Applicable if the current access permission is referenced by DiagnosticIOControl. The value of DcmDspDidControlRolebased is calculated based on the bits of all referenced DiagnosticAuth Role.bitPosition, where a value of 2^bitPosition is used for each DiagnosticAuthRole.bitPosition. The accumulated value represents a bitfield that has a bit set for each DiagnosticAuthRole.bit Position.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01143]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl
BSW Parameter	**BSW Type**
DcmDspDidControlSecurityLevelRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to DcmDspSecurityRow Security levels allowed to control this DID. If there is no reference, no check of security level shall be done.	
Template Description	
This represents the associated DiagnosticSecurityLevels	
M2 Parameter	
DiagnosticAccessPermission.securityLevel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00620]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl
BSW Parameter	**BSW Type**
DcmDspDidControlSessionRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to DcmDspSessionRow Sessions allowed to control this DID. If there is no reference, no check of session level shall be done.	
Template Description	
This represents the associated DiagnosticSessions	
M2 Parameter	
DiagnosticAccessPermission.diagnosticSession	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00621]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl
BSW Parameter	**BSW Type**
DcmDspDidFreezeCurrentState	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This indicates the presence of "FreezeCurrentState".	
Template Description	
DiagnosticIOControl.freezeCurrentState: Setting this attribute to true represents the ability of the Dcm to execute a freezeCurrentState. **DiagnosticIoControlNeeds.freezeCurrentStateSupported:** This attribute determines, if the referenced port supports temporary freezing of I/O value.	
M2 Parameter	
DiagnosticIOControl.freezeCurrentState, DiagnosticIoControlNeeds.freezeCurrentStateSupported	
Mapping Rule	**Mapping Type**



△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00624]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl
BSW Parameter	**BSW Type**
DcmDspDidResetToDefault	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This indicates the presence of "ResetToDefault".	
Template Description	
DiagnosticIOControl.resetToDefault: Setting this attribute to true represents the ability of the Dcm to execute a resetToDefault. **DiagnosticIoControlNeeds.resetToDefaultSupported:** This represents a flag for the existence of the ResetToDefault operation in the service interface.	
M2 Parameter	
DiagnosticIOControl.resetToDefault, DiagnosticIoControlNeeds.resetToDefaultSupported	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00623]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl
BSW Parameter	**BSW Type**
DcmDspDidShortTermAdjustment	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This indicates the presence of "ShortTermAdjustment".	
Template Description	
DiagnosticIOControl.shortTermAdjustment: Setting this attribute to true represents the ability of the Dcm to execute a shortTermAdjustment. **DiagnosticIoControlNeeds.shortTermAdjustmentSupported:** This attribute determines, if the referenced port supports temporarily setting of I/O value to a specific value provided by the diagnostic tester.	
M2 Parameter	
DiagnosticIOControl.shortTermAdjustment, DiagnosticIoControlNeeds. shortTermAdjustmentSupported	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00625]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo
BSW Parameter	**BSW Type**

▽

△

DcmDspDidDynamicallyDefined	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Indicates if this DID can be dynamically defined true = DID can be dynamically defined false = DID can not be dynamically defined	
Template Description	
DiagnosticDataIdentifier: This meta-class represents the ability to model a diagnostic data identifier (DID) that is fully specified regarding the payload at configuration-time. **DiagnosticDynamicDataIdentifier:** This meta-class represents the ability to define a diagnostic data identifier (DID) at run-time.	
M2 Parameter	
DiagnosticDataIdentifier, DiagnosticDynamicDataIdentifier	
Mapping Rule	**Mapping Type**
true: in case the DiagnosticAbstractDataIdentifier for the DID value is aggregated by Diagnostic DynamicDataIdentifier false: in case the DiagnosticAbstractDataIdentifier for the DID value is aggregated by DiagnosticDataIdentifier	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00612]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidRead
BSW Parameter	**BSW Type**
DcmDspDidReadRoleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to DcmDspAuthenticationRow that defines a role in that this DID can be read.	
Template Description	
This attribute allows for the specification of the position of the enclosing role in a bitfield of roles.	
M2 Parameter	
DiagnosticAuthRole.bitPosition	
Mapping Rule	**Mapping Type**
Applicable if the current access permission is referenced by DiagnosticReadDataByIdentifier. The value of DcmDspDidReadRole is calculated based calculated based on the bits of all referenced DiagnosticAuthRole.bitPosition, where a value of 2^bitPosition is used for each DiagnosticAuth Role.bitPosition. The accumulated value represents a bitfield that has a bit set for each Diagnostic AuthRole.bitPosition.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01141]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidRead
BSW Parameter	**BSW Type**
DcmDspDidReadSecurityLevelRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to DcmDspSecurityRow Referenced security levels are allowed to read this DID. If there is no reference, no check of security level shall be done.	
Template Description	
This represents the associated DiagnosticSecurityLevels	

▽

△

M2 Parameter	
DiagnosticAccessPermission.securityLevel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00614]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidRead
BSW Parameter	**BSW Type**
DcmDspDidReadSessionRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to DcmDspSessionRow Referenced sessions are allowed to read this DID. If there is no reference, no check of session level shall be done.	
Template Description	
This represents the associated DiagnosticSessions	
M2 Parameter	
DiagnosticAccessPermission.diagnosticSession	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00615]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidWrite
BSW Parameter	**BSW Type**
DcmDspDidWriteRoleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to DcmDspAuthenticationRow that defines a role in that this DID can be written.	
Template Description	
This attribute allows for the specification of the position of the enclosing role in a bitfield of roles.	
M2 Parameter	
DiagnosticAuthRole.bitPosition	
Mapping Rule	**Mapping Type**
Applicable if the current access permission is referenced by DiagnosticWriteDataByIdentifier. The value of DcmDspDidWriteRolecalculated is calculated based based on the bits of all referenced DiagnosticAuthRole.bitPosition, where a value of 2^bitPositionis used for each DiagnosticAuth Role.bitPosition. The accumulated value represents a bitfield that has a bit set for each Diagnostic AuthRole.bitPosition.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01142]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidWrite
BSW Parameter	**BSW Type**

▽

△

DcmDspDidWriteSecurityLevelRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to DcmDspSecurityRow Referenced security levels are allowed to wirte this DID. If there is no reference, no check of security level shall be done.	
Template Description	
This represents the associated DiagnosticSecurityLevels	
M2 Parameter	
DiagnosticAccessPermission.securityLevel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00617]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidWrite
BSW Parameter	**BSW Type**
DcmDspDidWriteSessionRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to DcmDspSessionRow Referenced sessions are allowed to write this DID. If there is no reference, no check of session level shall be done.	
Template Description	
This represents the associated DiagnosticSessions	
M2 Parameter	
DiagnosticAccessPermission.diagnosticSession	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00618]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspEcuReset/DcmDspEcuResetRow
BSW Parameter	**BSW Type**
DcmDspEcuResetId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the subfunction ID	
Template Description	
This represents the maximum number of source elements of the dynamically created DID.	
M2 Parameter	
DiagnosticDynamicallyDefineDataIdentifier.maxSourceElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01113]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspEcuReset/DcmDspEcuResetRow
BSW Parameter	**BSW Type**
DcmResponseToEcuReset	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the answer to EcuReset service should come: Before or after the reset.	
Template Description	
This attribute defines whether the response to the EcuReset service shall be transmitted before or after the actual reset.	
M2 Parameter	
DiagnosticEcuResetClass.respondToReset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01039]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspEcuReset/DcmDspEcuResetRow/DcmResponseToEcuReset
BSW Parameter	**BSW Type**
AFTER_RESET	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Answer to EcuReset service should come after the reset.	
Template Description	
Answer to EcuReset service should come after the reset.	
M2 Parameter	
DiagnosticResponseToEcuResetEnum.respondAfterReset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspEcuReset/DcmDspEcuResetRow/DcmResponseToEcuReset
BSW Parameter	**BSW Type**
BEFORE_RESET	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Answer to EcuReset service should come before the reset.	
Template Description	
Answer to EcuReset service should come before the reset.	
M2 Parameter	
DiagnosticResponseToEcuResetEnum.respondBeforeReset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp
BSW Parameter	**BSW Type**
DcmDspMaxDidToRead	ECUC-INTEGER-PARAM-DEF
BSW Description	
Indicates the maximum number of allowed DIDs in a single 'ReadDataByIdentifier' request for non-OBD DID requests (at least one DID outside the OBD range of 0xF400-0xF8FF).	
Template Description	
This attribute represents the maximum number of allowed DIDs in a single instance of DiagnosticReadDataByIdentifier for non-OBD DID requests (at least one DID outside the OBD range of 0xF400-0xF8FF).	
M2 Parameter	
DiagnosticReadDataByIdentifierClass.maxDidToRead	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00638]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp
BSW Parameter	**BSW Type**
DcmDspMaxPeriodicDidToRead	ECUC-INTEGER-PARAM-DEF
BSW Description	
Indicates the maximum allowed periodicDIDs which can be read in a single "ReadDataByPeriodicIdentifier" request.	
Template Description	
This represents the maximum number of data identifiers that can be included in one request.	
M2 Parameter	
DiagnosticReadDataByPeriodicIDClass.maxPeriodicDidToRead	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00956]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo
BSW Parameter	**BSW Type**
DcmDspMemoryIdValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Value of the memory device identifier used. Each DcmDspMemoryIdInfo should have a unique ID. The MemoryIdValue is retrieved from the request messages (RMBA,WMBA,RD,RU,DDDI) according to ISO-14229-1 with the most significant byte of the request parameter memoryAddress.	
Template Description	
This represents the identification of the memory segment.	
M2 Parameter	
DiagnosticMemoryIdentifier.id	



△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00913]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemory RangeByLabelInfo
BSW Parameter	**BSW Type**
DcmDspReadMemoryRangeByLabelHigh	ECUC-STRING-PARAM-DEF
BSW Description	
High memory address as label (string) of a range allowed for reading.	
Template Description	
This represents a symbolic label for the upper bound for addresses of the memory segment.	
M2 Parameter	
DiagnosticMemoryIdentifier.memoryHighAddressLabel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01070]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemory RangeByLabelInfo
BSW Parameter	**BSW Type**
DcmDspReadMemoryRangeByLabelLow	ECUC-STRING-PARAM-DEF
BSW Description	
Low memory address as label (string) of a range allowed for reading.	
Template Description	
This represents a symbolic label for the lower bound for addresses of the memory segment.	
M2 Parameter	
DiagnosticMemoryIdentifier.memoryLowAddressLabel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01069]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemory RangeByLabelInfo
BSW Parameter	**BSW Type**
DcmDspReadMemoryRangeSecurityLevelRef	ECUC-REFERENCE-DEF
BSW Description	
Link to the Security Access Levels needed for read access on this memory address. If there is no reference, no check of security level shall be done.	

▽

△

Template Description	
This represents the associated DiagnosticSecurityLevels	
M2 Parameter	
DiagnosticAccessPermission.securityLevel	
Mapping Rule	**Mapping Type**
DiagnosticMemoryIdentifier referenced in the role memoryRange is referenced by a Diagnostic ReadMemoryByAddress The accessPermission holds the security level information.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01071]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeByLabelInfo
BSW Parameter	**BSW Type**
DcmDspReadMemoryRangeSessionLevelRef	ECUC-REFERENCE-DEF
BSW Description	
Link to the session level needed for access to this memory address range. If there is no reference, no check of session level shall be done.	
Template Description	
This represents the associated DiagnosticSessions	
M2 Parameter	
DiagnosticAccessPermission.diagnosticSession	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01088]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeInfo
BSW Parameter	**BSW Type**
DcmDspReadMemoryRangeHigh	ECUC-INTEGER-PARAM-DEF
BSW Description	
High memory address of a range allowed for reading	
Template Description	
This represents the upper bound for addresses of the memory segment.	
M2 Parameter	
DiagnosticMemoryIdentifier.memoryHighAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00787]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeInfo
BSW Parameter	**BSW Type**
DcmDspReadMemoryRangeLow	ECUC-INTEGER-PARAM-DEF
BSW Description	
Low memory address of a range allowed for reading	
Template Description	
This represents the lower bound for addresses of the memory segment.	
M2 Parameter	
DiagnosticMemoryIdentifier.memoryLowAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00786]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeInfo
BSW Parameter	**BSW Type**
DcmDspReadMemoryRangeSecurityLevelRef	ECUC-REFERENCE-DEF
BSW Description	
Link to the Security Access Levels needed for read access on this memory address. If there is no reference, no check of security level shall be done.	
Template Description	
This represents the associated DiagnosticSecurityLevels	
M2 Parameter	
DiagnosticAccessPermission.securityLevel	
Mapping Rule	**Mapping Type**
DiagnosticMemoryIdentifier referenced in the role memoryRange is referenced by a Diagnostic ReadMemoryByAddress The accessPermission holds the security level information.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00788]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeInfo
BSW Parameter	**BSW Type**
DcmDspReadMemoryRangeSessionLevelRef	ECUC-REFERENCE-DEF
BSW Description	
Link to the session level needed for access to this memory address range. If there is no reference, no check of session level shall be done.	
Template Description	
This represents the associated DiagnosticSessions	
M2 Parameter	
DiagnosticAccessPermission.diagnosticSession	



△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01086]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeByLabelInfo
BSW Parameter	**BSW Type**
DcmDspWriteMemoryRangeByLabelHigh	ECUC-STRING-PARAM-DEF
BSW Description	
High memory address as label (string) of a range allowed for writing.	
Template Description	
This represents a symbolic label for the upper bound for addresses of the memory segment.	
M2 Parameter	
DiagnosticMemoryIdentifier.memoryHighAddressLabel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01075]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeByLabelInfo
BSW Parameter	**BSW Type**
DcmDspWriteMemoryRangeByLabelLow	ECUC-STRING-PARAM-DEF
BSW Description	
Low memory address as label (string) of a range allowed for writing.	
Template Description	
This represents a symbolic label for the lower bound for addresses of the memory segment.	
M2 Parameter	
DiagnosticMemoryIdentifier.memoryLowAddressLabel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01074]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeByLabelInfo
BSW Parameter	**BSW Type**
DcmDspWriteMemoryRangeSecurityLevelRef	ECUC-REFERENCE-DEF
BSW Description	
Link to the Security Access Levels needed for write access on this memory address. If there is no reference, no check of security level shall be done.	

▽

△

Template Description	
This represents the associated DiagnosticSecurityLevels	
M2 Parameter	
DiagnosticAccessPermission.securityLevel	
Mapping Rule	**Mapping Type**
DiagnosticMemoryIdentifier referenced in the role memoryRange is referenced by a DiagnosticWriteMemoryByAddress The accessPermission holds the security level information.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01076]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeByLabelInfo
BSW Parameter	**BSW Type**
DcmDspWriteMemoryRangeSessionLevelRef	ECUC-REFERENCE-DEF
BSW Description	
Link to the session level needed for access to this memory address range. If there is no reference, no check of session level shall be done.	
Template Description	
This represents the associated DiagnosticSessions	
M2 Parameter	
DiagnosticAccessPermission.diagnosticSession	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01089]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeInfo
BSW Parameter	**BSW Type**
DcmDspWriteMemoryRangeHigh	ECUC-INTEGER-PARAM-DEF
BSW Description	
High memory address of a range allowed for writing.	
Template Description	
This represents the upper bound for addresses of the memory segment.	
M2 Parameter	
DiagnosticMemoryIdentifier.memoryHighAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00791]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeInfo

BSW Parameter	BSW Type
DcmDspWriteMemoryRangeLow	ECUC-INTEGER-PARAM-DEF

BSW Description
Low memory address of a range allowed for writing

Template Description
This represents the lower bound for addresses of the memory segment.

M2 Parameter
DiagnosticMemoryIdentifier.memoryLowAddress

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00790]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeInfo

BSW Parameter	BSW Type
DcmDspWriteMemoryRangeSecurityLevelRef	ECUC-REFERENCE-DEF

BSW Description
Link to the Security Access Levels needed for write access on this memory address. If there is no reference, no check of security level shall be done.

Template Description
This represents the associated DiagnosticSecurityLevels

M2 Parameter
DiagnosticAccessPermission.securityLevel

Mapping Rule	Mapping Type
DiagnosticMemoryIdentifier referenced in the role memoryRange is referenced by a Diagnostic WriteMemoryByAddress The accessPermission holds the security level information.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00793]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeInfo

BSW Parameter	BSW Type
DcmDspWriteMemoryRangeSessionLevelRef	ECUC-REFERENCE-DEF

BSW Description
Link to the session level needed for access to this memory address range. If there is no reference, no check of session level shall be done.

Template Description
This represents the associated DiagnosticSessions

M2 Parameter
DiagnosticAccessPermission.diagnosticSession

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01087]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemoryTransfer/DcmDspMemoryTransferIdInfo
BSW Parameter	**BSW Type**
DcmDspMemoryIdValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Value of the memory device identifier used. Each DcmDspMemoryIdInfo should have a unique ID. The MemoryIdValue is retrieved from the request messages (RMBA,WMBA,RD,RU,DDDI) according to ISO-14229-1 with the most significant byte of the request parameter memoryAddress.	
Template Description	
This represents the identification of the memory segment.	
M2 Parameter	
DiagnosticMemoryIdentifier.id	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01138]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPeriodicTransmission
BSW Parameter	**BSW Type**
DcmDspMaxPeriodicDidScheduler	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the maximum number of periodicDataIdentifiers that can be scheduled concurrently.	
Template Description	
This represents the maximum number of periodic data identifiers that can be scheduled in parallel.	
M2 Parameter	
DiagnosticReadDataByPeriodicIDClass.schedulerMaxNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00962]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPeriodicTransmission
BSW Parameter	**BSW Type**
DcmDspPeriodicTransmissionFastRate	ECUC-FLOAT-PARAM-DEF
BSW Description	

▽

△

This parameter give the transmission rate of the requested periodicDataIdentifiers to be used if the parameter transmission Mode given in the ReadDataByPeriodicID request is equal to 0x03 ("sendAtFastRate"). This parameter value in seconds have to be configured as a multiple of DcmTaskTime. min: A negative value and zero is not allowed.
Template Description
DiagnosticPeriodicRate.period: This represents the period of the DiagnosticPeriodicRate in seconds. **DiagnosticPeriodicRate.periodicRateCategory:** This attribute represents the category of the periodic rate.
M2 Parameter
DiagnosticPeriodicRate.period, DiagnosticPeriodicRate.periodicRateCategory

Mapping Rule	Mapping Type
The parameter shall exist if DiagnosticPeriodicRate.periodicRateCategory is set to DiagnosticPeriodicRateCategoryEnum.periodicRateFast.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00960]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPeriodicTransmission

BSW Parameter	BSW Type
DcmDspPeriodicTransmissionMediumRate	ECUC-FLOAT-PARAM-DEF

BSW Description
This parameter give the transmission rate of the requested periodicDataIdentifiers to be used if the parameter transmission Mode given in the ReadDataByPeriodicID request is equal to 0x02 ("sendAtMediumRate"). This parameter value in seconds have to be configured as a multiple of DcmTaskTime. min: A negative value and zero is not allowed.
Template Description
DiagnosticPeriodicRate.period: This represents the period of the DiagnosticPeriodicRate in seconds. **DiagnosticPeriodicRate.periodicRateCategory:** This attribute represents the category of the periodic rate.
M2 Parameter
DiagnosticPeriodicRate.period, DiagnosticPeriodicRate.periodicRateCategory

Mapping Rule	Mapping Type
The parameter shall exist if DiagnosticPeriodicRate.periodicRateCategory is set to DiagnosticPeriodicRateCategoryEnum.periodicRateMedium.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00959]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPeriodicTransmission

BSW Parameter	BSW Type
DcmDspPeriodicTransmissionSlowRate	ECUC-FLOAT-PARAM-DEF

BSW Description

▽

△

This parameter give the transmission rate of the requested periodicDataIdentifiers to be used if the parameter transmission Mode given in the ReadDataByPeriodicID request is equal to 0x01 ("sendAtSlowRate"). This parameter value in seconds have to be configured as a multiple of DcmTaskTime. min: A negative value and zero is not allowed.
Template Description
DiagnosticPeriodicRate.period: This represents the period of the DiagnosticPeriodicRate in seconds. **DiagnosticPeriodicRate.periodicRateCategory:** This attribute represents the category of the periodic rate.
M2 Parameter
DiagnosticPeriodicRate.period, DiagnosticPeriodicRate.periodicRateCategory

Mapping Rule	Mapping Type
The parameter shall exist if DiagnosticPeriodicRate.periodicRateCategory is set to DiagnosticPeriodicRateCategoryEnum.periodicRateSlow.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00958]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp

BSW Parameter	BSW Type
DcmDspPid	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container defines the availability of a PID to the DCM.	
Template Description	
This meta-class represents the ability to model a diagnostic parameter identifier (PID) for the purpose of executing on-board diagnostics (OBD).	
M2 Parameter	
DiagnosticParameterIdentifier	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00626]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid

BSW Parameter	BSW Type
DcmDspPidData	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container defines the parameter for a Signal in the PID.	
Template Description	
This represents the data carried by the DiagnosticParameterIdentifier.	
M2 Parameter	
DiagnosticParameterIdentifier.dataElement	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00865]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData
BSW Parameter	**BSW Type**
DcmDspPidByteOffset	ECUC-INTEGER-PARAM-DEF
BSW Description	
This is the position in bytes of the PID structure and will not start at position 0 in case a support information is available (for packeted PIDs).	
Template Description	
This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticDataIdentifier, DiagnosticParameterIdentifier, or DiagnosticRoutineSubfunction.	
M2 Parameter	
DiagnosticParameter.bitOffset	
Mapping Rule	**Mapping Type**
bitOffset / 8	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01107]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData
BSW Parameter	**BSW Type**
DcmDspPidDataByteSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the array length in bytes or the maximum array length for variable datalengths.	
Template Description	
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
S/R via array: DcmDspPidDataByteSize= maxNumberOfElements * (baseTypeSize / 8) C/S of FNC callback: DcmDspPidDataByteSize= maxNumberOfElements Note: 8 is the baseType Size of UINT8	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01108]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData
BSW Parameter	**BSW Type**
DcmDspPidDataSupportInfo	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container defines the supported information.	
Template Description	
This attribute represents the ability to define which bit of the support info byte is representing this part of the PID.	



△

M2 Parameter	
DiagnosticParameter.supportInfo	
Mapping Rule	**Mapping Type**
Applicable if the DiagnosticParameter is owned by a DiagnosticParameterIdentifier	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00874]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidDataSupportInfo
BSW Parameter	**BSW Type**
DcmDspPidDataSupportInfoBit	ECUC-INTEGER-PARAM-DEF
BSW Description	
Referenced Bit of the SupportInfo	
Template Description	
defines the bit in the SupportInfo byte, which represents the PID DataElement pidSize / position / size. Unit: byte.	
M2 Parameter	
DiagnosticParameterSupportInfo.supportInfoBit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00876]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidDataSupportInfo
BSW Parameter	**BSW Type**
DcmDspPidDataSupportInfoRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to DcmDspPidSupportInfo	
Template Description	
This represents the supported information associated with the DiagnosticParameterIdentifier.	
M2 Parameter	
DiagnosticParameterIdentifier.supportInfoByte	
Mapping Rule	**Mapping Type**
Shall refer to the DiagnosticParameterIdentifier.supportInfoByte of the enclosing Diagnostic ParameterIdentifier	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00875]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDsp DiagnosisScaling/DcmDspAlternativeDataType
BSW Parameter	**BSW Type**
DcmDspTextTableMapping	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.
Template Description
This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.
M2 Parameter
CompuMethod

Mapping Rule	Mapping Type
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00999]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspDiagnosisScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping

BSW Parameter	BSW Type
DcmDspDiagnosisRepresentationDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The data value in the diagnosis representation.	
Template Description	
This represents a textual constant in the computation method.	
M2 Parameter	
CompuConstTextContent.vt	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01001]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspDiagnosisScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping

BSW Parameter	BSW Type
DcmDspInternalDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The ECU internal data value.	
Template Description	
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit:** This specifies the upper limit of a of the scale.	

▽

△

M2 Parameter	
CompuScale.lowerLimit, CompuScale.upperLimit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01000]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataEndianness
BSW Parameter	**BSW Type**
BIG_ENDIAN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Most significant byte shall be stored at the lowest address.	
Template Description	
This attribute specifies the byte order of the base type.	
M2 Parameter	
BaseTypeDirectDefinition.byteOrder	
Mapping Rule	**Mapping Type**
BaseTypeDirectDefinition.byteOrder == ByteOrderEnum.mostSignificantByteFirst	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataEndianness
BSW Parameter	**BSW Type**
LITTLE_ENDIAN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Most significant byte shall be stored at the highest address.	
Template Description	
This attribute specifies the byte order of the base type.	
M2 Parameter	
BaseTypeDirectDefinition.byteOrder	
Mapping Rule	**Mapping Type**
BaseTypeDirectDefinition.byteOrder ==ByteOrderEnum.mostSignificantByteLast	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataEndianness
BSW Parameter	**BSW Type**
OPAQUE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Opaque data endianness	
Template Description	
This attribute specifies the byte order of the base type.	
M2 Parameter	
BaseTypeDirectDefinition.byteOrder	
Mapping Rule	**Mapping Type**
BaseTypeDirectDefinition.byteOrder == ByteOrderEnum.opaque	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01
BSW Parameter	**BSW Type**
DcmDspPidDataReadFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Function name for reading PID data value. This is only relevant if DcmDspPidDataUsePort==USE_DATA_SYNCH_FNC. This parameter is related to the interface Xxx_ReadData.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
The BswServiceDependency should have aRoleBasedBswModuleEntryAssignment that in turn has attribute role set to xxx_ReadData and points to a BswModuleEntry.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00629]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataType
BSW Parameter	**BSW Type**
BOOLEAN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is boolean.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement. • baseTypeEncoding = BOOLEAN • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataType
BSW Parameter	**BSW Type**
DOUBLE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is double (64 bit float).	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataType

▽

△

BSW Parameter	BSW Type
DOUBLE_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is double (64 bit float) array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`	
Mapping Rule	**Mapping Type**
• `baseTypeEncoding` = IEEE754 • `baseTypeSize` = 64 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01001) • `arraySizeSemantics` either does not exist or exists and is set to `fixedSize` (cf. TPS_DEXT_01001) Derivation from `fixedLength` =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataType

BSW Parameter	BSW Type
SINT16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement. • baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataType
BSW Parameter	**BSW Type**
SINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeSize, BaseTypeDirectDefinition.baseTypeEncoding, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataType
BSW Parameter	**BSW Type**
SINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint32.	

▽

△

Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement. • baseTypeEncoding = 2C • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataType

BSW Parameter	BSW Type
SINT32_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeSize, BaseTypeDirectDefinition.baseTypeEncoding, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	

Mapping Rule	Mapping Type
• baseTypeEncoding = 2C • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataType

BSW Parameter	BSW Type
SINT64	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint64.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = NONE • `baseTypeSize` = 64 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist Derivation from `fixedLength` =1 possible.	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataType

BSW Parameter	BSW Type
SINT64_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint64 array.

Template Description

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
• baseTypeEncoding = 2C • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType

BSW Parameter	BSW Type
SINT8	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement. • baseTypeEncoding = 2C • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataType
BSW Parameter	**BSW Type**
SINT8_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeSize, BaseTypeDirectDefinition.baseTypeEncoding, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = 2C • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixed Size (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataType
BSW Parameter	**BSW Type**
UINT16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint16.	
Template Description	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement. • baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataType

BSW Parameter	**BSW Type**
UINT16_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
Type of the data is uint16 array.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE,UTF-16 • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataType

BSW Parameter	BSW Type
UINT32	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is uint32.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
referenced by `swDataDefProps` of the `DiagnosticParameter` with the role `DiagnosticParameterIdentifier.dataElement`. • `baseTypeEncoding` = NONE,UTF-32 • `baseTypeSize` = 32 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataType

BSW Parameter	BSW Type
UINT32_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is uint32 array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`

▽

△

Mapping Rule	Mapping Type
• baseTypeEncoding = NONE • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType
BSW Parameter	**BSW Type**
UINT64	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint64.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	

Mapping Rule	Mapping Type
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType
BSW Parameter	**BSW Type**
UINT64_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Type of the data is uint64 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataType
BSW Parameter	**BSW Type**
UINT8	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement. • baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataType
BSW Parameter	**BSW Type**
UINT8_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE,WINDOWS-1252,UTF-8 • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataUsePort
BSW Parameter	**BSW Type**
USE_DATA_SENDER_RECEIVER	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The DCM will access the Data using an Port requiring a SenderReceiverInterface.	
Template Description	

▽

△

This represents the ability to define a mapping of a diagnostic service to a software-component. This kind of service mapping is applicable for the usage of SenderReceiverInterfaces or event/notifier semantics in Service Interfaces on the adaptive platform.	
M2 Parameter	
DiagnosticServiceDataMapping	
Mapping Rule	**Mapping Type**
DiagnosticServiceDataMapping.diagnosticDataElement refers to a DiagnosticDataElement that in turn is aggregated in the PID definition where the PortPrototype referenced in the role mapped DataElement.contextPort refers to a DataInterface where attribute isService is set to false.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataUsePort
BSW Parameter	**BSW Type**
USE_DATA_SENDER_RECEIVER_AS_SERVICE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The DCM will access the Data using an service Port requiring a SenderReceiverInterface (with isService=true).	
Template Description	
This represents the ability to define a mapping of a diagnostic service to a software-component. This kind of service mapping is applicable for the usage of SenderReceiverInterfaces or event/notifier semantics in Service Interfaces on the adaptive platform.	
M2 Parameter	
DiagnosticServiceDataMapping	
Mapping Rule	**Mapping Type**
DiagnosticServiceDataMapping.diagnosticDataElement refers to a DiagnosticDataElement that in turn is aggregated in the PID definition where the PortPrototype referenced in the role mapped DataElement.contextPort refers to a DataInterface where attribute isService is set to true.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataUsePort
BSW Parameter	**BSW Type**
USE_DATA_SYNCH_CLIENT_SERVER	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The DCM will access the Data using an R-Port requiring a synchronous ClientServerInterface.	
Template Description	
This represents the ability to define a mapping of a diagnostic service to a software-component or a basic-software module. If the former is used then this kind of service mapping is applicable for the usage of ClientServerInterfaces.	
M2 Parameter	
DiagnosticServiceSwMapping	
Mapping Rule	**Mapping Type**
DiagnosticServiceSwMapping.diagnosticDataElement refers to a DiagnosticDataElement that in turn is aggregated in the PID definition. DiagnosticServiceSwMapping.mappedSwcService Dependency shall exist.	full

▽

△

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid DataUsePort

BSW Parameter	BSW Type
USE_DATA_SYNCH_FNC	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

The DCM will access the Data using the functions that are defined in parameters of type EcucFunctionNameDef (but without DcmDspDataReadDataLengthFnc) in the DcmDspData container. DCM_E_PENDING return value is not allowed and Op Status parameter is not existing in the prototype.

Template Description

This represents the ability to define a mapping of a diagnostic service to a software-component or a basic-software module. If the former is used then this kind of service mapping is applicable for the usage of ClientServerInterfaces.

M2 Parameter

DiagnosticServiceSwMapping

Mapping Rule	Mapping Type
DiagnosticServiceSwMapping.diagnosticDataElement refers to a DiagnosticDataElement that in turn is aggregated in the PID definition and that also defines the role mappedBswService Dependency.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid Service01ExternalSRDataElementClass/DcmDataElementInstance

BSW Parameter	BSW Type
DcmDataElementInstanceRef	ECUC-INSTANCE-REFERENCE-DEF

BSW Description

Instance Reference to the primitive or array data which shall be read or written. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationPrimitiveDataType of category VALUE or BOOLEAN or ApplicationArrayDataType or if the AutosarDataPrototype is typed with a ImplementationDataType of category VALUE, ARRAY or TYPE_REFERENCE that in turn boils down to VALUE or ARRAY

Template Description

DiagnosticServiceDataMapping.mappedDataElement:
This represents the dataElement in the application software that is accessed for diagnostic purpose. This role is applicable on the classic platform.

DiagnosticEnvDataElementCondition:
This meta-class represents the ability to formulate a diagnostic environment condition based on the value of a data element owned by the application software.

M2 Parameter

DiagnosticServiceDataMapping.mappedDataElement, DiagnosticEnvDataElementCondition

Mapping Rule	Mapping Type
DiagnosticServiceDataMapping maps to a primitive data.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00991]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidService01ExternalSRDataElementClass/DcmSubElementInDataElementInstance

BSW Parameter	BSW Type
DcmSubElementInDataElementInstanceRef	ECUC-INSTANCE-REFERENCE-DEF

BSW Description

Instance Reference to the primitve sub-element (at any level) of composite data in a port which shall be read. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationCompositeDataType.

Template Description

This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.

M2 Parameter

`DiagnosticServiceSwMapping.mappedBswServiceDependency`

Mapping Rule	Mapping Type
DiagnosticServiceDataMapping maps to a primitive element within a composite data, where the AutosarDataPrototype is typed with a ApplicationCompositeDataType.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00990]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService02

BSW Parameter	BSW Type
DcmDspPidDataDemRef	ECUC-REFERENCE-DEF

BSW Description

Reference to DemPidDataElement in DEM configuration. Allows to link the DCM PID and DEM PID configuration for Mode $02.

Template Description

This represents the PID associated with this instance of the OBD mode 0x02 service.

M2 Parameter

`DiagnosticPowertrainFreezeFrame.pid`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00887]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid

BSW Parameter	BSW Type
DcmDspPidIdentifier	ECUC-INTEGER-PARAM-DEF

BSW Description

1 byte Identifier of the PID

Within each DcmConfigSet all DcmDspPidIdentifier values shall be unique.

Template Description

This is the numerical identifier used to identify the DiagnosticParameterIdentifier in the scope of diagnostic workflow (see SAE J1979-DA).



△

M2 Parameter	
DiagnosticParameterIdentifier.id	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00627]

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidService	
BSW Parameter		**BSW Type**
DCM_SERVICE_01		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
A PID is used with service $01 only.		
Template Description		
This represents the PID associated with this instance of the OBD mode 0x01 service.		
M2 Parameter		
DiagnosticRequestCurrentPowertrainData.pid		
Mapping Rule		**Mapping Type**
Applicable if the DiagnosticParameterIdentifier is only referenced by a DiagnosticRequestCurrent PowertrainData		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidService	
BSW Parameter		**BSW Type**
DCM_SERVICE_01_02		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
A PID is used with service $01 and $02. Allowed with a PID configuration containing data elements on byte basis.		
Template Description		
DiagnosticRequestCurrentPowertrainData.pid: This represents the PID associated with this instance of the OBD mode 0x01 service. **DiagnosticPowertrainFreezeFrame.pid:** This represents the PID associated with this instance of the OBD mode 0x02 service.		
M2 Parameter		
DiagnosticRequestCurrentPowertrainData.pid, DiagnosticPowertrainFreezeFrame.pid		
Mapping Rule		**Mapping Type**
Applicable if the DiagnosticParameterIdentifier is referenced by both a DiagnosticRequestCurrent PowertrainData and a DiagnosticPowertrainFreezeFrame		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidService	
BSW Parameter		**BSW Type**

▽

△

DCM_SERVICE_02	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
A PID is used with service $02 only. Allowed with a PID configuration containing data elements on byte basis.	
Template Description	
This represents the PID associated with this instance of the OBD mode 0x02 service.	
M2 Parameter	
DiagnosticPowertrainFreezeFrame.pid	
Mapping Rule	**Mapping Type**
Applicable if the DiagnosticParameterIdentifier is only referenced by a DiagnosticPowertrainFreeze Frame	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid
BSW Parameter	**BSW Type**
DcmDspPidSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Length of a PID in byte(s).	
Template Description	
The size of the entire PID can be greater than the sum of the data elements because padding might be applied. Unit: byte.	
M2 Parameter	
DiagnosticParameterIdentifier.pidSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00870]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid
BSW Parameter	**BSW Type**
DcmDspPidSupportInfo	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container defines the support information (typically byte A) to declare the usability of the data bytes within the so-called packeted PIDs (e.g. PID$68).	
Template Description	
This represents the supported information associated with the DiagnosticParameterIdentifier.	
M2 Parameter	
DiagnosticParameterIdentifier.supportInfoByte	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00871]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidSupportInfo
BSW Parameter	**BSW Type**
DcmDspPidSupportInfoLen	ECUC-INTEGER-PARAM-DEF
BSW Description	
Length of the support information in bytes.	
Template Description	
This represents the size of the supportInfo within the PID. Unit: byte.	
M2 Parameter	
DiagnosticSupportInfoByte.size	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00873]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidSupportInfo
BSW Parameter	**BSW Type**
DcmDspPidSupportInfoPos	ECUC-INTEGER-PARAM-DEF
BSW Description	
Position of the support information in bytes.	
Template Description	
This represents the position of the supportInfo in the PID. Unit: byte.	
M2 Parameter	
DiagnosticSupportInfoByte.position	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00872]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspReadDTCInformation
BSW Parameter	**BSW Type**
DcmDspReadDTCInformationSupportedObdUdsDtc Separation	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If this parameter is set to true, the OBD UDS DTC separation is enabled. If this parameter is not configured or set to false, the OBD UDS DTC separation is disabled.	
Template Description	
3 Byte OBD DTC value based on the definition from SAE J2012. The existence of this attribute is only required if separated UDS and OBD DTC values are used for SAE J1979-2. If this attribute does not exist, then UDS DTC values are used with J1979-2.	
M2 Parameter	
DiagnosticTroubleCodeUds.obdDtcValue3Byte	
Mapping Rule	**Mapping Type**



△

DcmDspReadDTCInformationSupportedObdUdsDtcSeparation shall exist and set to true if, in the context of an enclosing DiagnosticContributionSet, at least one DiagnosticTroubleCodeObd exists where attribute obdDTCValue3Byte exists, independently of its value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01214]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspReadDTCInformation/DcmDspReadDTCInformationUserDefinedFaultMemory
BSW Parameter	**BSW Type**
DcmDspReadDTCInformationUserDefinedFaultMemoryRoleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to DcmDspAuthenticationRow that defines a role in that this user defined memory can be accessed.	
Template Description	
This attribute allows for the specification of the position of the enclosing role in a bitfield of roles.	
M2 Parameter	
DiagnosticAuthRole.bitPosition	
Mapping Rule	**Mapping Type**
Applicable for DiagnosticAuthRoles refererenced by a DiagnosticMemoryDestination with DiagnosticMemoryDestination.memoryId having the same value as DcmDspReadDTCInformationUserDefinedFaultMemoryId. The value of DiagnosticAccessPermission is calculated based on the bits of all referenced DiagnosticAuthRole.bitPosition, where a value of 2^bitPosition is used for each DiagnosticAuthRole.bitPosition. The accumulated value represents a bitfield that has a bit set for each DiagnosticAuthRole.bitPosition.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01150]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp
BSW Parameter	**BSW Type**
DcmDspRequestControl	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration (parameters) of the "Request control of on-board system, test or component" service (Service $08). The DCM will request the control using an R-Port requiring a PortInteface RequestControlServices_{Tid}. The R-Port is named RequestControlServices_{Tid} where {Tid} is the name of the container DcmDspRequestControl.	
Template Description	
This meta-class represents the ability to model an instance of the OBD mode 0x08 service.	
M2 Parameter	
DiagnosticRequestControlOfOnBoardDevice	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00637]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRequestControl
BSW Parameter	**BSW Type**
DcmDspRequestControlInBufferSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Number of bytes to be provided in the input buffer of the interface RequestControlServices_{Tid} for OBD Service $08	
Template Description	
This represents the specified data size for the request message. Unit: byte.	
M2 Parameter	
DiagnosticTestRoutineIdentifier.requestDataSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00722]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRequestControl
BSW Parameter	**BSW Type**
DcmDspRequestControlOutBufferSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Number of bytes to be provided in the output buffer of the interface RequestControlServices_{Tid} for OBD Service $08	
Template Description	
This represents the specified data size for the response message. Unit:byte.	
M2 Parameter	
DiagnosticTestRoutineIdentifier.responseDataSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00723]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRequestControl
BSW Parameter	**BSW Type**
DcmDspRequestControlTestId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Test Id for Service $08	
Template Description	
This represents the numerical id of the DiagnosticTestIdentifier (see SAE J1979-DA).	
M2 Parameter	
DiagnosticTestRoutineIdentifier.id	
Mapping Rule	**Mapping Type**
The value shall be taken from DiagnosticRequestControlOfOnBoardDevice.testId.id if available.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00656]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp
BSW Parameter	**BSW Type**
DcmDspRequestFileTransfer	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration for RequestFileTransfer. This container only exists if RequestFileTransfer is configured.	
Template Description	
This diagnostic service instance implements the UDS service 0x38.	
M2 Parameter	
DiagnosticRequestFileTransfer	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01034]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe
BSW Parameter	**BSW Type**
DcmDspRoeMaxNumChangeOfDataIdentfierEvents	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of events that can be simultaneously configured with SubFunction onChangeOfDataIdentifier.	
Template Description	
The maximum number of events that can be simultaneously configured with sub function onChangeOfDataIdentifier.	
M2 Parameter	
DiagnosticResponseOnEventClass.maxNumChangeOfDataIdentfierEvents	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01236]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe
BSW Parameter	**BSW Type**
DcmDspRoeMaxNumComparisonOfValueEvents	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of events that can be simultaneously configured with SubFunction onComparisonOfValues.	
Template Description	
The maximum number of events that can be simultaneously configured with sub function onComparisonOfValues.	
M2 Parameter	
DiagnosticResponseOnEventClass.maxNumComparisionOfValueEvents	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01237]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe
BSW Parameter	**BSW Type**
DcmDspRoeMaxNumOfStoredDTCStatusChangedEvents	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of changed DTC events that can be stored within one DcmDspRoeSchedulerRate interval.	
Template Description	
The maximum number of DTCs that can be stored as DTCs with change status within one ResponseOnEventSchedulerRate interval.	
M2 Parameter	
DiagnosticResponseOnEventClass.maxNumberOfStoredDTCStatusChangedEvents	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01239]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe
BSW Parameter	**BSW Type**
DcmDspRoeMaxSupportedDIDLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum DID response size in bytes allowed for a DID that is used for onComparisonOfValues or onChangeOfDataIdentifier.	
Template Description	
The maximum number of measurable data bytes allowed for each DID that is used for comparison or data change.	
M2 Parameter	
DiagnosticResponseOnEventClass.maxSupportedDIDLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01238]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe
BSW Parameter	**BSW Type**
DcmDspRoeSchedulerRate	ECUC-FLOAT-PARAM-DEF
BSW Description	
Calling rate of the RoE scheduler. The parameter value is defined in seconds and shall be a multiple of DcmTaskTime.	
Template Description	
The call rate of the periodic scheduler to compare the values of the DataIdentifier (DID) or to detect DTC status changes.	
M2 Parameter	
DiagnosticResponseOnEventClass.responseOnEventSchedulerRate	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01235]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe
BSW Parameter	**BSW Type**
DcmDspRoeStoreEventSupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Configures if ResponseOnEvent storageState is supported. If set to true, the use of the storageState bit is supported and if set to false the use of the storageState bit is not supported.	
Template Description	
Specifies if the storeEvent functionality of the ResponseOnEvent diagnostic service shall be supported or not. If set to true, the storeEvent functionality is available. If set to false the storeEvent functionality is not available.	
M2 Parameter	
DiagnosticResponseOnEventClass.storeEventEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01234]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp
BSW Parameter	**BSW Type**
DcmDspRoutine	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration (parameters) for Routines	
Template Description	
This meta-class represents the ability to define a diagnostic routine.	
M2 Parameter	
DiagnosticRoutine	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00640]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine
BSW Parameter	**BSW Type**
DcmDspRequestRoutineResults	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Provides the configuration of RequestResult subservice for RoutineControl service. Existence indicates that the Request RoutineResults in the RoutineControl is supported.	
Template Description	
DiagnosticRoutine.requestResult: This represents the ability to request the result of a running routine. **DiagnosticRoutineNeeds.diagRoutineType:** This denotes the type of diagnostic routine which is implemented by the referenced server port.	
M2 Parameter	
DiagnosticRoutine.requestResult, DiagnosticRoutineNeeds.diagRoutineType	



△

Mapping Rule	Mapping Type
1:1 mapping for DiagnosticRoutine.requestResult OR DiagnosticRoutineNeeds.diagRoutineTyoe == asynchronous	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01023]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults
BSW Parameter	**BSW Type**
DcmDspRequestRoutineResultsFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Function name for request to application the results of a routine. (Routine_RequestResults-function) This parameter is related to the interface Xxx_RequestResults.	
Template Description	
Specialization of ServiceDependency in the context of an BswInternalBehavior. It allows to associate BswModuleEntries and data defined for a BSW module or cluster to a given ServiceNeeds element.	
M2 Parameter	
BswServiceDependency	
Mapping Rule	**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00753]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults
BSW Parameter	**BSW Type**
DcmDspRequestRoutineResultsIn	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Provide description of input parameter of RequestResult subservice for RoutineControl service.	
Template Description	
This represents the request parameters.	
M2 Parameter	
DiagnosticRequestRoutineResults.request	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01116]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn
BSW Parameter	**BSW Type**
DcmDspRequestRoutineResultsInSignal	ECUC-PARAM-CONF-CONTAINER-DEF

▽

△

BSW Description	
Provides description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataIn_n elements in the XXX_RequestResult function call.	
Template Description	
This represents the related dataElement of the DiagnosticParameter	
M2 Parameter	
DiagnosticAbstractParameter.dataElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01117]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType
BSW Parameter	**BSW Type**
DcmDspTextTableMapping	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.	
Template Description	
This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.	
M2 Parameter	
CompuMethod	
Mapping Rule	**Mapping Type**
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00999]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspDiagnosisRepresentationDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The data value in the diagnosis representation.	

▽

△

Template Description	
This represents a textual constant in the computation method.	
M2 Parameter	
CompuConstTextContent.vt	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01001]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspInternalDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The ECU internal data value.	
Template Description	
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit:** This specifies the upper limit of a of the scale.	
M2 Parameter	
CompuScale.lowerLimit, CompuScale.upperLimit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01000]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal
BSW Parameter	**BSW Type**
DcmDspRoutineParameterSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Provide the size of a RoutineControl parameter in bytes	
Template Description	
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01119]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal

BSW Parameter	BSW Type
DcmDspRoutineSignalEndianness	ECUC-ENUMERATION-PARAM-DEF

BSW Description
Defines the endianness of the data.

Template Description
This attribute specifies the byte order of the base type.

M2 Parameter
BaseTypeDirectDefinition.byteOrder

Mapping Rule	Mapping Type
• If the parent container is DcmDspRequestRoutineResultsOutSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response • If the parent container is DcmDspStopRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request • If the parent container is DcmDspStopRoutineOutSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response • If the parent container is DcmDspStartRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request • If the parent container is DcmDspStartRoutineOutSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response • If the parent container is DcmDspRequestRoutineResultsInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response	full

Mapping Status	ECUC Parameter ID
valid	[ECUC_Dcm_01013]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal

BSW Parameter	BSW Type
DcmDspRoutineSignalPos	ECUC-INTEGER-PARAM-DEF

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

Template Description
This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticDataIdentifier, DiagnosticParameterIdentifier, or DiagnosticRoutineSubfunction.

M2 Parameter
DiagnosticAbstractParameter.bitOffset

Mapping Rule	Mapping Type
1:1 mapping	full

Mapping Status	ECUC Parameter ID
valid	[ECUC_Dcm_01118]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignalCompositePool
BSW Parameter	**BSW Type**
DcmDspRequestRoutineResultsInSignal	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Provides description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataIn_n elements in the XXX_RequestResult function call.	
Template Description	
This represents the related dataElement of the DiagnosticParameter	
M2 Parameter	
DiagnosticAbstractParameter.dataElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01117]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignalCompositePool/DcmDspRequestRoutineResultsInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType
BSW Parameter	**BSW Type**
DcmDspTextTableMapping	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.	
Template Description	
This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.	
M2 Parameter	
CompuMethod	
Mapping Rule	**Mapping Type**
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00999]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignalCompositePool/DcmDspRequestRoutineResultsInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspDiagnosisRepresentationDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The data value in the diagnosis representation.	
Template Description	
This represents a textual constant in the computation method.	
M2 Parameter	
`CompuConstTextContent.vt`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01001]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignalCompositePool/DcmDspRequestRoutineResultsInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspInternalDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The ECU internal data value.	
Template Description	
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit:** This specifies the upper limit of a of the scale.	
M2 Parameter	
`CompuScale.lowerLimit`, `CompuScale.upperLimit`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01000]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignalCompositePool/DcmDspRequestRoutineResultsInSignal
BSW Parameter	**BSW Type**
DcmDspRoutineParameterSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Provide the size of a RoutineControl parameter in bytes	
Template Description	



△

The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01119]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignalCompositePool/DcmDspRequestRoutineResultsInSignal
BSW Parameter	**BSW Type**
DcmDspRoutineSignalEndianness	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the endianness of the data.	
Template Description	
This attribute specifies the byte order of the base type.	
M2 Parameter	
BaseTypeDirectDefinition.byteOrder	
Mapping Rule	**Mapping Type**
• If the parent container is DcmDspRequestRoutineResultsOutSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response • If the parent container is DcmDspStopRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request • If the parent container is DcmDspStopRoutineOutSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response • If the parent container is DcmDspStartRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request • If the parent container is DcmDspStartRoutineOutSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response • If the parent container is DcmDspRequestRoutineResultsInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01013]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignalCompositePool/DcmDspRequestRoutineResultsInSignal
BSW Parameter	**BSW Type**
DcmDspRoutineSignalPos	ECUC-INTEGER-PARAM-DEF

▽

△

BSW Description	
Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.	
Template Description	
This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticDataIdentifier, DiagnosticParameterIdentifier, or DiagnosticRoutineSubfunction.	
M2 Parameter	
DiagnosticAbstractParameter.bitOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01118]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults
BSW Parameter	**BSW Type**
DcmDspRequestRoutineResultsOut	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Provide description of output parameter of RequestResult subservice for RoutineControl service.	
Template Description	
This represents the response parameters.	
M2 Parameter	
DiagnosticRequestRoutineResults.response	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00831]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut
BSW Parameter	**BSW Type**
DcmDspRequestRoutineResultsOutSignal	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Provides description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataOutN elements in the XXX_RequestResult function call.	
Template Description	
This represents the related dataElement of the DiagnosticParameter	
M2 Parameter	
DiagnosticAbstractParameter.dataElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00836]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType

BSW Parameter	BSW Type
DcmDspTextTableMapping	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists).

In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.

Template Description

This meta-class represents the ability to express the relationship between a physical value and the mathematical representation.

Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.

M2 Parameter

`CompuMethod`

Mapping Rule	Mapping Type
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00999]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping

BSW Parameter	BSW Type
DcmDspDiagnosisRepresentationDataValue	ECUC-INTEGER-PARAM-DEF

BSW Description

The data value in the diagnosis representation.

Template Description

This represents a textual constant in the computation method.

M2 Parameter

`CompuConstTextContent.vt`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01001]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping

BSW Parameter	BSW Type

△

DcmDspInternalDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The ECU internal data value.	
Template Description	
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit:** This specifies the upper limit of a of the scale.	
M2 Parameter	
`CompuScale.lowerLimit, CompuScale.upperLimit`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01000]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal
BSW Parameter	**BSW Type**
DcmDspRoutineParameterSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Provide the size of a RoutineControl parameter in bytes	
Template Description	
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
`DiagnosticDataElement.maxNumberOfElements`	
Mapping Rule	**Mapping Type**
Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00838]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal
BSW Parameter	**BSW Type**
DcmDspRoutineSignalEndianness	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the endianness of the data.	
Template Description	
This attribute specifies the byte order of the base type.	
M2 Parameter	
`BaseTypeDirectDefinition.byteOrder`	
Mapping Rule	**Mapping Type**

▽

△

• If the parent container is DcmDspRequestRoutineResultsOutSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response • If the parent container is DcmDspStopRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.request • If the parent container is DcmDspStopRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.response • If the parent container is DcmDspStartRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.request • If the parent container is DcmDspStartRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.response • If the parent container is DcmDspRequestRoutineResultsInSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01013]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequest RoutineResultsOut/DcmDspRequestRoutineResultsOutSignal
BSW Parameter	**BSW Type**
DcmDspRoutineSignalPos	ECUC-INTEGER-PARAM-DEF
BSW Description	
Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.	
Template Description	
This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticDataIdentifier, DiagnosticParameterIdentifier, or DiagnosticRoutineSubfunction.	
M2 Parameter	
`DiagnosticAbstractParameter.bitOffset`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00837]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequest RoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
BOOLEAN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is boolean.	
Template Description	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = BOOLEAN • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
DOUBLE	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is double (64 bit float).

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type

▽

△

• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
DOUBLE_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is double (64 bit float) array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
FLOAT	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Type of the data is float.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
FLOAT_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is float array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint16 array.	
Template Description	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics, DataPrototype.swDataDefProps

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize	

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT32_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is sint32 array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
referenced by `swDataDefProps` of the `DiagnosticParameter` with the role `DiagnosticRequestRoutineResults.response`. • `baseTypeEncoding` = 2C* `baseTypeSize` = 32 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01001) • `arraySizeSemantics` either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT64	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint64.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.



△

M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT64_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint64 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = 2C • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT8	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is sint8.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DataPrototype.swDataDefProps`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
referenced by `swDataDefProps` of the `DiagnosticParameter` with the role `DiagnosticRequestRoutineResults.response`. • `baseTypeEncoding` = 2C • `baseTypeSize` = 8 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT8_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is sint8 array.

Template Description

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics, DataPrototype.swDataDefProps

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C* baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
UINT16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Type of the signal is uint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE, UTF-32 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT32_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT64	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint64.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT64_N	ECUC-ENUMERATION-LITERAL-DEF

▽

△

BSW Description
Type of the data is uint64 array.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.
M2 Parameter
`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = NONE • `baseTypeSize` = 64 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01001) • `arraySizeSemantics` either does not exist or exists and is set to `fixedSize` (cf. TPS_DEXT_01001) Derivation from `fixedLength` =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
UINT8	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE,WINDOWS-1252,UTF-8 • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT8_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
rreferenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
VARIABLE_LENGTH	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is uint8[DcmDspRoutineParameterSize].

This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
referenced by `swDataDefProps` of the `DiagnosticParameter` with the role `DiagnosticRequestRoutineResults.response`. • `baseTypeEncoding` = NONE • `baseTypeSize` = 8 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01002) • `arraySizeSemantics` exists and is set to `ArraySizeSemanticsEnum.variableSize` (cf. TPS_DEXT_01002)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool

BSW Parameter	BSW Type
DcmDspRequestRoutineResultsOutSignal	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

Provides description of a routine signal used in RoutineControl service.

The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataOutN elements in the XXX_RequestResult function call.

Template Description

This represents the related dataElement of the DiagnosticParameter

M2 Parameter

`DiagnosticAbstractParameter.dataElement`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00836]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType
BSW Parameter	**BSW Type**
DcmDspTextTableMapping	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.	
Template Description	
This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.	
M2 Parameter	
CompuMethod	
Mapping Rule	**Mapping Type**
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00999]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspDiagnosisRepresentationDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The data value in the diagnosis representation.	
Template Description	
This represents a textual constant in the computation method.	
M2 Parameter	
CompuConstTextContent.vt	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01001]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping

BSW Parameter	BSW Type
DcmDspInternalDataValue	ECUC-INTEGER-PARAM-DEF

BSW Description
The ECU internal data value.

Template Description
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit:** This specifies the upper limit of a of the scale.

M2 Parameter
`CompuScale.lowerLimit`, `CompuScale.upperLimit`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01000]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal

BSW Parameter	BSW Type
DcmDspRoutineParameterSize	ECUC-INTEGER-PARAM-DEF

BSW Description
Provide the size of a RoutineControl parameter in bytes

Template Description
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter
`DiagnosticDataElement.maxNumberOfElements`

Mapping Rule	Mapping Type
Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00838]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal

BSW Parameter	BSW Type
DcmDspRoutineSignalEndianness	ECUC-ENUMERATION-PARAM-DEF

BSW Description
Defines the endianness of the data.



△

Template Description	
This attribute specifies the byte order of the base type.	
M2 Parameter	
BaseTypeDirectDefinition.byteOrder	
Mapping Rule	**Mapping Type**
• If the parent container is DcmDspRequestRoutineResultsOutSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response • If the parent container is DcmDspStopRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.request • If the parent container is DcmDspStopRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.response • If the parent container is DcmDspStartRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.request • If the parent container is DcmDspStartRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.response • If the parent container is DcmDspRequestRoutineResultsInSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01013]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequest RoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequest RoutineResultsOutSignal
BSW Parameter	**BSW Type**
DcmDspRoutineSignalPos	ECUC-INTEGER-PARAM-DEF
BSW Description	
Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.	
Template Description	
This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticDataIdentifier, DiagnosticParameterIdentifier, or DiagnosticRoutineSubfunction.	
M2 Parameter	
DiagnosticAbstractParameter.bitOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00837]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequest RoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequest RoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
BOOLEAN	ECUC-ENUMERATION-LITERAL-DEF

▽

△

BSW Description
Type of the signal is boolean.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = BOOLEAN • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
DOUBLE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is double (64 bit float).	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
DOUBLE_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is double (64 bit float) array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
FLOAT	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is float.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = IEEE754 • `baseTypeSize` = 32 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist Derivation from `fixedLength` =1 possible.	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
FLOAT_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is float array.

Template Description

▽

△

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = IEEE754 • `baseTypeSize` = 32 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01001) • `arraySizeSemantics` either does not exist or exists and is set to `fixedSize` (cf. TPS_DEXT_01001) Derivation from `fixedLength` =1 possible.	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT16	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is sint16.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DataPrototype.swDataDefProps`

Mapping Rule	Mapping Type

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**

▽

△

SINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT32_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C* baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT64	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint64.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = NONE • `baseTypeSize` = 64 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist Derivation from `fixedLength` =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT64_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint64 array.

Template Description

▽

△

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = 2C • `baseTypeSize` = 64 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01001) • `arraySizeSemantics` either does not exist or exists and is set to `fixedSize` (cf. TPS_DEXT_01001) Derivation from `fixedLength` =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT8	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is sint8.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DataPrototype.swDataDefProps`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT8_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics, DataPrototype.swDataDefProps	

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C* baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16	ECUC-ENUMERATION-LITERAL-DEF

▽

△

BSW Description
Type of the signal is uint16.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
UINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE, UTF-32 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT32_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Type of the signal is uint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT64	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint64.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT64_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint64 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
UINT8	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is uint8.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DataPrototype.swDataDefProps`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`

Mapping Rule	Mapping Type
referenced by `swDataDefProps` of the `DiagnosticParameter` with the role `DiagnosticRequestRoutineResults.response`. • `baseTypeEncoding` = NONE,WINDOWS-1252,UTF-8 • `baseTypeSize` = 8 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
UINT8_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is uint8 array.

Template Description

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps

Mapping Rule	Mapping Type
rreferenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignalCompositePool/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
VARIABLE_LENGTH	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
Type of the signal is uint8[DcmDspRoutineParameterSize]. This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics

Mapping Rule	Mapping Type

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01002) • arraySizeSemantics exists and is set to ArraySizeSemanticsEnum.variableSize (cf. TPS_DEXT_01002)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults
BSW Parameter	**BSW Type**
DcmDspRequestRoutineResultsRoleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to DcmDspAuthenticationRow that defines a role in that this routine results can be read.	
Template Description	
This attribute allows for the specification of the position of the enclosing role in a bitfield of roles.	
M2 Parameter	
DiagnosticAuthRole.bitPosition	
Mapping Rule	**Mapping Type**
Applicable if the current access permission is referenced by DiagnosticRequestRoutineResults. The value of DcmDspRequestRoutineResultsRole is calculated based the bits of all referenced DiagnosticAuthRole.bitPosition, where a value of 2^bitPosition is used for each DiagnosticAuth Role.bitPosition. The accumulated value represents a bitfield that has a bit set for each Diagnostic AuthRole.bitPosition.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01146]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine
BSW Parameter	**BSW Type**
DcmDspRoutineIdentifier	ECUC-INTEGER-PARAM-DEF
BSW Description	
2 bytes Identifier of the RID Within each DcmConfigSet all DcmDspRoutineIdentifier values shall be unique.	
Template Description	
This is the numerical identifier used to identify the DiagnosticRoutine in the scope of diagnostic workflow	
M2 Parameter	
DiagnosticRoutine.id	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00641]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine
BSW Parameter	**BSW Type**
DcmDspRoutineUsePort	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
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 Dcm DspRequestResultsRoutineFnc. If this is false, the DCM expects to find the names of the functions to be used in DcmDsp StartRoutineFnc, DcmDspStopRoutineFnc or DcmDspRequestResultsRoutineFnc.	
Template Description	
This represents the ability to define a mapping of a diagnostic service to a software-component or a basic-software module. If the former is used then this kind of service mapping is applicable for the usage of ClientServerInterfaces.	
M2 Parameter	
DiagnosticServiceSwMapping	
Mapping Rule	**Mapping Type**
TRUE: DiagnosticServiceSwMapping is having a SwcServiceDependency FALSE: DiagnosticServiceSwMapping is having a BswServiceDependency	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00724]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine
BSW Parameter	**BSW Type**
DcmDspStartRoutine	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Provides the configuration of Start subservice for RoutineControl service.	
Template Description	
DiagnosticRoutine.start: This represents the ability to start a routine **DiagnosticRoutineNeeds:** Specifies the general needs on the configuration of the Diagnostic Communication Manager (Dcm) which are not related to a particular item (e.g. a PID). The main use case is the mapping of service ports to the Dcm which are not related to a particular item.	
M2 Parameter	
DiagnosticRoutine.start, DiagnosticRoutineNeeds	
Mapping Rule	**Mapping Type**
A routine always comes with a start routine, independently of whether the execution is done synchronously or asynchronously.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01021]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine
BSW Parameter	**BSW Type**
DcmDspStartRoutineFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	

▽

△

Function name for request to application to start a routine. (Routine_Start-function) This parameter is related to the interface Xxx_Start.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00664]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine
BSW Parameter	**BSW Type**
DcmDspStartRoutineIn	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Provide description of input parameter of Start subservice for RoutineControl service	
Template Description	
This represents the request parameters.	
M2 Parameter	
DiagnosticStartRoutine.request	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00834]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn
BSW Parameter	**BSW Type**
DcmDspStartRoutineInSignal	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Provide description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataInN elements in the XXX_Start function call.	
Template Description	
This represents the related dataElement of the DiagnosticParameter	
M2 Parameter	
DiagnosticAbstractParameter.dataElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00845]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType

BSW Parameter	BSW Type
DcmDspTextTableMapping	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists).

In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.

Template Description

This meta-class represents the ability to express the relationship between a physical value and the mathematical representation.

Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.

M2 Parameter

`CompuMethod`

Mapping Rule	Mapping Type
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00999]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping

BSW Parameter	BSW Type
DcmDspDiagnosisRepresentationDataValue	ECUC-INTEGER-PARAM-DEF

BSW Description

The data value in the diagnosis representation.

Template Description

This represents a textual constant in the computation method.

M2 Parameter

`CompuConstTextContent.vt`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01001]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping

BSW Parameter	BSW Type
DcmDspInternalDataValue	ECUC-INTEGER-PARAM-DEF



△

BSW Description	
The ECU internal data value.	
Template Description	
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit:** This specifies the upper limit of a of the scale.	
M2 Parameter	
`CompuScale.lowerLimit, CompuScale.upperLimit`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01000]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal
BSW Parameter	**BSW Type**
DcmDspRoutineParameterSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Provide the size of a RoutineControl parameter in bytes	
Template Description	
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
`DiagnosticDataElement.maxNumberOfElements`	
Mapping Rule	**Mapping Type**
"Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00847]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal
BSW Parameter	**BSW Type**
DcmDspRoutineSignalEndianness	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the endianness of the data.	
Template Description	
This attribute specifies the byte order of the base type.	
M2 Parameter	
`BaseTypeDirectDefinition.byteOrder`	
Mapping Rule	**Mapping Type**

▽

△

• If the parent container is DcmDspRequestRoutineResultsOutSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response • If the parent container is DcmDspStopRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.request • If the parent container is DcmDspStopRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.response • If the parent container is DcmDspStartRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.request • If the parent container is DcmDspStartRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.response • If the parent container is DcmDspRequestRoutineResultsInSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01013]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal

BSW Parameter	**BSW Type**
DcmDspRoutineSignalPos	ECUC-INTEGER-PARAM-DEF
BSW Description	
Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.	
Template Description	
This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticDataIdentifier, DiagnosticParameterIdentifier, or DiagnosticRoutineSubfunction.	
M2 Parameter	
DiagnosticAbstractParameter.bitOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00846]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
BOOLEAN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is boolean.	
Template Description	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = BOOLEAN • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
DOUBLE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is double (64 bit float).	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**

▽

△

• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
DOUBLE_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is double (64 bit float) array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
FLOAT	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Type of the data is float.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
FLOAT_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is float array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint16 array.	
Template Description	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics, DataPrototype.swDataDefProps

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize	

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT32_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is sint32 array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
referenced by `swDataDefProps` of the `DiagnosticParameter` with the role `DiagnosticRequestRoutineResults.response`. • `baseTypeEncoding` = 2C* `baseTypeSize` = 32 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01001) • `arraySizeSemantics` either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT64	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint64.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.



△

M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT64_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint64 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = 2C • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT8	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is sint8.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DataPrototype.swDataDefProps`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
referenced by `swDataDefProps` of the `DiagnosticParameter` with the role `DiagnosticRequestRoutineResults.response`. • `baseTypeEncoding` = 2C • `baseTypeSize` = 8 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT8_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is sint8 array.

Template Description

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics, DataPrototype.swDataDefProps

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C* baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
UINT16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Type of the signal is uint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE, UTF-32 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT32_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT64	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint64.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT64_N	ECUC-ENUMERATION-LITERAL-DEF

▽

△

BSW Description
Type of the data is uint64 array.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.
M2 Parameter
`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = NONE • `baseTypeSize` = 64 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01001) • `arraySizeSemantics` either does not exist or exists and is set to `fixedSize` (cf. TPS_DEXT_01001) Derivation from `fixedLength` =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
UINT8	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
Type of the signal is uint8.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE,WINDOWS-1252,UTF-8 • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT8_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
rreferenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
VARIABLE_LENGTH	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is uint8[DcmDspRoutineParameterSize].

This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
referenced by `swDataDefProps` of the `DiagnosticParameter` with the role `DiagnosticRequestRoutineResults.response`. • `baseTypeEncoding` = NONE • `baseTypeSize` = 8 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01002) • `arraySizeSemantics` exists and is set to `ArraySizeSemanticsEnum.variableSize` (cf. TPS_DEXT_01002)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool

BSW Parameter	BSW Type
DcmDspStartRoutineInSignal	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

Provide description of a routine signal used in RoutineControl service.

The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataInN elements in the XXX_Start function call.

Template Description

This represents the related dataElement of the DiagnosticParameter

M2 Parameter

`DiagnosticAbstractParameter.dataElement`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00845]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType

BSW Parameter	BSW Type
DcmDspTextTableMapping	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists).

In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.

Template Description

This meta-class represents the ability to express the relationship between a physical value and the mathematical representation.

Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.

M2 Parameter

`CompuMethod`

Mapping Rule	Mapping Type
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00999]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping

BSW Parameter	BSW Type
DcmDspDiagnosisRepresentationDataValue	ECUC-INTEGER-PARAM-DEF

BSW Description

The data value in the diagnosis representation.

Template Description

This represents a textual constant in the computation method.

M2 Parameter

`CompuConstTextContent.vt`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01001]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping

BSW Parameter	BSW Type

△

DcmDspInternalDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The ECU internal data value.	
Template Description	
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit:** This specifies the upper limit of a of the scale.	
M2 Parameter	
`CompuScale.lowerLimit, CompuScale.upperLimit`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01000]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal
BSW Parameter	**BSW Type**
DcmDspRoutineParameterSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Provide the size of a RoutineControl parameter in bytes	
Template Description	
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
`DiagnosticDataElement.maxNumberOfElements`	
Mapping Rule	**Mapping Type**
"Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00847]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal
BSW Parameter	**BSW Type**
DcmDspRoutineSignalEndianness	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the endianness of the data.	
Template Description	
This attribute specifies the byte order of the base type.	
M2 Parameter	
`BaseTypeDirectDefinition.byteOrder`	
Mapping Rule	**Mapping Type**

▽

△

• If the parent container is DcmDspRequestRoutineResultsOutSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response • If the parent container is DcmDspStopRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.request • If the parent container is DcmDspStopRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.response • If the parent container is DcmDspStartRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.request • If the parent container is DcmDspStartRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.response • If the parent container is DcmDspRequestRoutineResultsInSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01013]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal
BSW Parameter	**BSW Type**
DcmDspRoutineSignalPos	ECUC-INTEGER-PARAM-DEF
BSW Description	
Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.	
Template Description	
This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticDataIdentifier, DiagnosticParameterIdentifier, or DiagnosticRoutineSubfunction.	
M2 Parameter	
`DiagnosticAbstractParameter.bitOffset`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00846]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
BOOLEAN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is boolean.	
Template Description	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = BOOLEAN • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
DOUBLE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is double (64 bit float).	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**

▽

△

• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
DOUBLE_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is double (64 bit float) array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
FLOAT	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Type of the data is float.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
FLOAT_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is float array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint16 array.	
Template Description	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics, DataPrototype.swDataDefProps

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize	

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT32_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is sint32 array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
referenced by `swDataDefProps` of the `DiagnosticParameter` with the role `DiagnosticRequestRoutineResults.response`. • `baseTypeEncoding` = 2C* `baseTypeSize` = 32 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01001) • `arraySizeSemantics` either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT64	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint64.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

▽

△

M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT64_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint64 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = 2C • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT8	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is sint8.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DataPrototype.swDataDefProps`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
referenced by `swDataDefProps` of the `DiagnosticParameter` with the role `DiagnosticRequestRoutineResults.response`. • `baseTypeEncoding` = 2C • `baseTypeSize` = 8 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT8_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is sint8 array.

Template Description

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics, DataPrototype.swDataDefProps

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C* baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
UINT16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	

Mapping Rule	**Mapping Type**

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Type of the signal is uint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE, UTF-32 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT32_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT64	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint64.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT64_N	ECUC-ENUMERATION-LITERAL-DEF

▽

△

BSW Description
Type of the data is uint64 array.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.
M2 Parameter
`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = NONE • `baseTypeSize` = 64 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01001) • `arraySizeSemantics` either does not exist or exists and is set to `fixedSize` (cf. TPS_DEXT_01001) Derivation from `fixedLength` =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
UINT8	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE,WINDOWS-1252,UTF-8 • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT8_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
rreferenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignalCompositePool/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
VARIABLE_LENGTH	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is uint8[DcmDspRoutineParameterSize].

This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
referenced by `swDataDefProps` of the `DiagnosticParameter` with the role `DiagnosticRequestRoutineResults.response`. • `baseTypeEncoding` = NONE • `baseTypeSize` = 8 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01002) • `arraySizeSemantics` exists and is set to `ArraySizeSemanticsEnum.variableSize` (cf. TPS_DEXT_01002)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine

BSW Parameter	BSW Type
DcmDspStartRoutineOut	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

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

Template Description

This represents the response parameters.

M2 Parameter

`DiagnosticStartRoutine.response`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00835]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut
BSW Parameter	**BSW Type**
DcmDspStartRoutineOutSignal	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Provide description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataOutN elements in the XXX_Start function call.	
Template Description	
This represents the related dataElement of the DiagnosticParameter	
M2 Parameter	
DiagnosticAbstractParameter.dataElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00848]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/ DcmDspStartRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType
BSW Parameter	**BSW Type**
DcmDspTextTableMapping	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.	
Template Description	
This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.	
M2 Parameter	
CompuMethod	
Mapping Rule	**Mapping Type**
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00999]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/ DcmDspStartRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDsp TextTableMapping
BSW Parameter	**BSW Type**



△

DcmDspDiagnosisRepresentationDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The data value in the diagnosis representation.	
Template Description	
This represents a textual constant in the computation method.	
M2 Parameter	
`CompuConstTextContent.vt`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01001]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspInternalDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The ECU internal data value.	
Template Description	
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit:** This specifies the upper limit of a of the scale.	
M2 Parameter	
`CompuScale.lowerLimit`, `CompuScale.upperLimit`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01000]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal
BSW Parameter	**BSW Type**
DcmDspRoutineParameterSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Provide the size of a RoutineControl parameter in bytes	
Template Description	
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
`DiagnosticDataElement.maxNumberOfElements`	
Mapping Rule	**Mapping Type**
Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutine SignalLength = maxNumberOfElements * 8.	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_Dcm_00850]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal

BSW Parameter	BSW Type
DcmDspRoutineSignalEndianness	ECUC-ENUMERATION-PARAM-DEF

BSW Description

Defines the endianness of the data.

Template Description

This attribute specifies the byte order of the base type.

M2 Parameter

`BaseTypeDirectDefinition.byteOrder`

Mapping Rule	Mapping Type
• If the parent container is DcmDspRequestRoutineResultsOutSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response • If the parent container is DcmDspStopRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.request • If the parent container is DcmDspStopRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.response • If the parent container is DcmDspStartRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.request • If the parent container is DcmDspStartRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.response • If the parent container is DcmDspRequestRoutineResultsInSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response	full

Mapping Status	ECUC Parameter ID
valid	[ECUC_Dcm_01013]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal

BSW Parameter	BSW Type
DcmDspRoutineSignalPos	ECUC-INTEGER-PARAM-DEF

BSW Description

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

Template Description

This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticDataIdentifier, DiagnosticParameterIdentifier, or DiagnosticRoutineSubfunction.

M2 Parameter

`DiagnosticAbstractParameter.bitOffset`

Mapping Rule	Mapping Type

▽

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00867]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
BOOLEAN	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is boolean.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DataPrototype.swDataDefProps`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`

Mapping Rule	**Mapping Type**
referenced by `swDataDefProps` of the `DiagnosticParameter` with the role `DiagnosticRequestRoutineResults.response`. • `baseTypeEncoding` = BOOLEAN • `baseTypeSize` = 8 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
DOUBLE	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is double (64 bit float).

Template Description

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
DOUBLE_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is double (64 bit float) array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**

▽

△

• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
FLOAT	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is float.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
FLOAT_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Type of the data is float array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Type of the signal is sint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT32_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C* baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT64	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint64.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = NONE • `baseTypeSize` = 64 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist Derivation from `fixedLength` =1 possible.	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT64_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint64 array.

Template Description

▽

△

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = 2C • `baseTypeSize` = 64 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01001) • `arraySizeSemantics` either does not exist or exists and is set to `fixedSize` (cf. TPS_DEXT_01001) Derivation from `fixedLength` =1 possible.	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT8	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is sint8.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DataPrototype.swDataDefProps`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT8_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics, DataPrototype.swDataDefProps	

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C* baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Type of the signal is uint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
UINT32	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
Type of the signal is uint32.

Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.

M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE, UTF-32 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
UINT32_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
Type of the signal is uint32 array.

Template Description

▽

△

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
UINT64	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is uint64.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics

Mapping Rule	Mapping Type

▽

△

• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT64_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint64 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT8	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Type of the signal is uint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE,WINDOWS-1252,UTF-8 • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT8_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**

▽

△

rreferenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
VARIABLE_LENGTH	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint8[DcmDspRoutineParameterSize]. This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01002) • arraySizeSemantics exists and is set to ArraySizeSemanticsEnum.variableSize (cf. TPS_DEXT_01002)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool
BSW Parameter	**BSW Type**
DcmDspStartRoutineOutSignal	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

Provide description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataOutN elements in the XXX_Start function call.	
Template Description	
This represents the related dataElement of the DiagnosticParameter	
M2 Parameter	
DiagnosticAbstractParameter.dataElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00848]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType
BSW Parameter	**BSW Type**
DcmDspTextTableMapping	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.	
Template Description	
This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.	
M2 Parameter	
CompuMethod	
Mapping Rule	**Mapping Type**
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00999]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspDiagnosisRepresentationDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The data value in the diagnosis representation.	
Template Description	

▽

△

This represents a textual constant in the computation method.	
M2 Parameter	
CompuConstTextContent.vt	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01001]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspInternalDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The ECU internal data value.	
Template Description	
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit:** This specifies the upper limit of a of the scale.	
M2 Parameter	
CompuScale.lowerLimit, CompuScale.upperLimit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01000]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal
BSW Parameter	**BSW Type**
DcmDspRoutineParameterSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Provide the size of a RoutineControl parameter in bytes	
Template Description	
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00850]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal
BSW Parameter	**BSW Type**
DcmDspRoutineSignalEndianness	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the endianness of the data.	
Template Description	
This attribute specifies the byte order of the base type.	
M2 Parameter	
BaseTypeDirectDefinition.byteOrder	
Mapping Rule	**Mapping Type**
• If the parent container is DcmDspRequestRoutineResultsOutSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response • If the parent container is DcmDspStopRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.request • If the parent container is DcmDspStopRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.response • If the parent container is DcmDspStartRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.request • If the parent container is DcmDspStartRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.response • If the parent container is DcmDspRequestRoutineResultsInSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01013]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal
BSW Parameter	**BSW Type**
DcmDspRoutineSignalPos	ECUC-INTEGER-PARAM-DEF
BSW Description	
Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.	
Template Description	
This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticDataIdentifier, DiagnosticParameterIdentifier, or DiagnosticRoutineSubfunction.	
M2 Parameter	
DiagnosticAbstractParameter.bitOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00867]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
BOOLEAN	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is boolean.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DataPrototype.swDataDefProps`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`

Mapping Rule	Mapping Type
referenced by `swDataDefProps` of the `DiagnosticParameter` with the role `DiagnosticRequestRoutineResults.response`. • `baseTypeEncoding` = BOOLEAN • `baseTypeSize` = 8 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
DOUBLE	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is double (64 bit float).

Template Description

▽

△

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = IEEE754 • `baseTypeSize` = 64 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist Derivation from `fixedLength` =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
DOUBLE_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is double (64 bit float) array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`

▽

△

Mapping Rule	Mapping Type
• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
FLOAT	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is float.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics

Mapping Rule	Mapping Type
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

▽

△

BSW Parameter	BSW Type
FLOAT_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is float array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics, DataPrototype.swDataDefProps	

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**

▽

△

SINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT32_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C* baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT64	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint64.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = NONE • `baseTypeSize` = 64 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist Derivation from `fixedLength` =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT64_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint64 array.

Template Description

▽

△

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = 2C • `baseTypeSize` = 64 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01001) • `arraySizeSemantics` either does not exist or exists and is set to `fixedSize` (cf. TPS_DEXT_01001) Derivation from `fixedLength` =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT8	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is sint8.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DataPrototype.swDataDefProps`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT8_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics, DataPrototype.swDataDefProps	

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C* baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16	ECUC-ENUMERATION-LITERAL-DEF

▽

△

BSW Description	
Type of the signal is uint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
UINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE, UTF-32 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT32_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Type of the signal is uint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT64	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint64.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT64_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint64 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
UINT8	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is uint8.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DataPrototype.swDataDefProps`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`

Mapping Rule	Mapping Type
referenced by `swDataDefProps` of the `DiagnosticParameter` with the role `DiagnosticRequestRoutineResults.response`. • `baseTypeEncoding` = NONE,WINDOWS-1252,UTF-8 • `baseTypeSize` = 8 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
UINT8_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is uint8 array.

Template Description

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps

Mapping Rule	Mapping Type
rreferenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignalCompositePool/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
VARIABLE_LENGTH	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
Type of the signal is uint8[DcmDspRoutineParameterSize]. This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics

Mapping Rule	Mapping Type

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01002) • arraySizeSemantics exists and is set to ArraySizeSemanticsEnum.variableSize (cf. TPS_DEXT_01002)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine
BSW Parameter	**BSW Type**
DcmDspStartRoutineRoleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to DcmDspAuthenticationRow that defines a role in that this routine can be started.	
Template Description	
This attribute allows for the specification of the position of the enclosing role in a bitfield of roles.	
M2 Parameter	
DiagnosticAuthRole.bitPosition	
Mapping Rule	**Mapping Type**
Applicable if the current access permission is referenced by DiagnosticStartRoutine. The value of DcmDspStartRoutineRole is calculated based on the bits of all referenced DiagnosticAuthRole.bit Position, where a value of 2^bitPosition is used for each DiagnosticAuthRole.bitPosition. The accumulated value represents a bitfield that has a bit set for each DiagnosticAuthRole.bitPosition.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01144]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine
BSW Parameter	**BSW Type**
DcmDspStopRoutine	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Provides the configuration of Stop subservice for RoutineControl service. Existence indicates that the StopRoutine in the RoutineControl is supported.	
Template Description	
DiagnosticRoutine.stop: This represents the ability to stop a running routine. **DiagnosticRoutineNeeds.diagRoutineType:** This denotes the type of diagnostic routine which is implemented by the referenced server port.	
M2 Parameter	
DiagnosticRoutine.stop, DiagnosticRoutineNeeds.diagRoutineType	
Mapping Rule	**Mapping Type**
1:1 mapping for DiagnosticRoutine.stop OR DiagnosticRoutineNeeds.diagRoutineType == asynchronous	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01022]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine
BSW Parameter	**BSW Type**
DcmDspStopRoutineFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Function name for request to application to stop a routine. (Routine_Stop-function) This parameter is related to the interface Xxx_Stop.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00752]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine
BSW Parameter	**BSW Type**
DcmDspStopRoutineIn	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Provide description of input parameter of Stop subservice for RoutineControl service.	
Template Description	
This represents the request parameters.	
M2 Parameter	
DiagnosticStopRoutine.request	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00832]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn
BSW Parameter	**BSW Type**
DcmDspStopRoutineInSignal	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Provide description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataInN elements in the XXX_Stop function call.	
Template Description	
This represents the related dataElement of the DiagnosticParameter	
M2 Parameter	
DiagnosticAbstractParameter.dataElement	
Mapping Rule	**Mapping Type**



△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00839]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType

BSW Parameter	**BSW Type**
DcmDspTextTableMapping	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists).

In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.

Template Description

This meta-class represents the ability to express the relationship between a physical value and the mathematical representation.

Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.

M2 Parameter

`CompuMethod`

Mapping Rule	**Mapping Type**
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00999]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping

BSW Parameter	**BSW Type**
DcmDspDiagnosisRepresentationDataValue	ECUC-INTEGER-PARAM-DEF

BSW Description

The data value in the diagnosis representation.

Template Description

This represents a textual constant in the computation method.

M2 Parameter

`CompuConstTextContent.vt`

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01001]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping

BSW Parameter	BSW Type
DcmDspInternalDataValue	ECUC-INTEGER-PARAM-DEF

BSW Description
The ECU internal data value.

Template Description
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit:** This specifies the upper limit of a of the scale.

M2 Parameter
`CompuScale.lowerLimit`, `CompuScale.upperLimit`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01000]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal

BSW Parameter	BSW Type
DcmDspRoutineParameterSize	ECUC-INTEGER-PARAM-DEF

BSW Description
Provide the size of a RoutineControl parameter in bytes

Template Description
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter
`DiagnosticDataElement.maxNumberOfElements`

Mapping Rule	Mapping Type
Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00841]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal

BSW Parameter	BSW Type
DcmDspRoutineSignalEndianness	ECUC-ENUMERATION-PARAM-DEF

BSW Description
Defines the endianness of the data.

Template Description
This attribute specifies the byte order of the base type.

▽

△

M2 Parameter	
BaseTypeDirectDefinition.byteOrder	
Mapping Rule	**Mapping Type**
• If the parent container is DcmDspRequestRoutineResultsOutSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response • If the parent container is DcmDspStopRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.request • If the parent container is DcmDspStopRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.response • If the parent container is DcmDspStartRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.request • If the parent container is DcmDspStartRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.response • If the parent container is DcmDspRequestRoutineResultsInSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01013]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal
BSW Parameter	**BSW Type**
DcmDspRoutineSignalPos	ECUC-INTEGER-PARAM-DEF
BSW Description	
Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.	
Template Description	
This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticDataIdentifier, DiagnosticParameterIdentifier, or DiagnosticRoutineSubfunction.	
M2 Parameter	
DiagnosticAbstractParameter.bitOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00840]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignalCompositePool
BSW Parameter	**BSW Type**
DcmDspStopRoutineInSignal	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

Provide description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataInN elements in the XXX_Stop function call.	
Template Description	
This represents the related dataElement of the DiagnosticParameter	
M2 Parameter	
DiagnosticAbstractParameter.dataElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00839]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignalCompositePool/DcmDspStopRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType
BSW Parameter	**BSW Type**
DcmDspTextTableMapping	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.	
Template Description	
This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.	
M2 Parameter	
CompuMethod	
Mapping Rule	**Mapping Type**
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00999]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignalCompositePool/DcmDspStopRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspDiagnosisRepresentationDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The data value in the diagnosis representation.	
Template Description	

▽

△

This represents a textual constant in the computation method.	
M2 Parameter	
CompuConstTextContent.vt	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01001]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignalCompositePool/DcmDspStopRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspInternalDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The ECU internal data value.	
Template Description	
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit:** This specifies the upper limit of a of the scale.	
M2 Parameter	
CompuScale.lowerLimit, CompuScale.upperLimit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01000]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignalCompositePool/DcmDspStopRoutineInSignal
BSW Parameter	**BSW Type**
DcmDspRoutineParameterSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Provide the size of a RoutineControl parameter in bytes	
Template Description	
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00841]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignalCompositePool/DcmDspStopRoutineInSignal

BSW Parameter	BSW Type
DcmDspRoutineSignalEndianness	ECUC-ENUMERATION-PARAM-DEF

BSW Description

Defines the endianness of the data.

Template Description

This attribute specifies the byte order of the base type.

M2 Parameter

`BaseTypeDirectDefinition.byteOrder`

Mapping Rule	Mapping Type
• If the parent container is DcmDspRequestRoutineResultsOutSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response • If the parent container is DcmDspStopRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request • If the parent container is DcmDspStopRoutineOutSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response • If the parent container is DcmDspStartRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request • If the parent container is DcmDspStartRoutineOutSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response • If the parent container is DcmDspRequestRoutineResultsInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response	full

Mapping Status	ECUC Parameter ID
valid	[ECUC_Dcm_01013]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignalCompositePool/DcmDspStopRoutineInSignal

BSW Parameter	BSW Type
DcmDspRoutineSignalPos	ECUC-INTEGER-PARAM-DEF

BSW Description

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

Template Description

This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticDataIdentifier, DiagnosticParameterIdentifier, or DiagnosticRoutineSubfunction.

M2 Parameter

`DiagnosticAbstractParameter.bitOffset`

Mapping Rule	Mapping Type
1:1 mapping	full

Mapping Status	ECUC Parameter ID
valid	[ECUC_Dcm_00840]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine
BSW Parameter	**BSW Type**
DcmDspStopRoutineOut	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Provide description of output parameter of Stop subservice for RoutineControl service.	
Template Description	
This represents the response parameters.	
M2 Parameter	
DiagnosticStopRoutine.response	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00833]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut
BSW Parameter	**BSW Type**
DcmDspStopRoutineOutSignal	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Provide description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataOutN elements in the XXX_Stop function call.	
Template Description	
This represents the related dataElement of the DiagnosticParameter	
M2 Parameter	
DiagnosticAbstractParameter.dataElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00842]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/ DcmDspStopRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType
BSW Parameter	**BSW Type**
DcmDspTextTableMapping	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.	
Template Description	

▽

△

This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.	
M2 Parameter	
CompuMethod	
Mapping Rule	**Mapping Type**
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00999]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspDiagnosisRepresentationDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The data value in the diagnosis representation.	
Template Description	
This represents a textual constant in the computation method.	
M2 Parameter	
CompuConstTextContent.vt	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01001]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspInternalDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The ECU internal data value.	
Template Description	
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit:** This specifies the upper limit of a of the scale.	
M2 Parameter	
CompuScale.lowerLimit, CompuScale.upperLimit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01000]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal

BSW Parameter	BSW Type
DcmDspRoutineParameterSize	ECUC-INTEGER-PARAM-DEF

BSW Description
Provide the size of a RoutineControl parameter in bytes

Template Description
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter
DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
"Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00844]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal

BSW Parameter	BSW Type
DcmDspRoutineSignalEndianness	ECUC-ENUMERATION-PARAM-DEF

BSW Description
Defines the endianness of the data.

Template Description
This attribute specifies the byte order of the base type.

M2 Parameter
BaseTypeDirectDefinition.byteOrder

Mapping Rule	Mapping Type
• If the parent container is DcmDspRequestRoutineResultsOutSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response • If the parent container is DcmDspStopRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.request • If the parent container is DcmDspStopRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.response • If the parent container is DcmDspStartRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.request • If the parent container is DcmDspStartRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.response • If the parent container is DcmDspRequestRoutineResultsInSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01013]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal

BSW Parameter	BSW Type
DcmDspRoutineSignalPos	ECUC-INTEGER-PARAM-DEF

BSW Description

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

Template Description

This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticDataIdentifier, DiagnosticParameterIdentifier, or DiagnosticRoutineSubfunction.

M2 Parameter

`DiagnosticAbstractParameter.bitOffset`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00843]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
BOOLEAN	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is boolean.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DataPrototype.swDataDefProps`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`

Mapping Rule	Mapping Type
referenced by `swDataDefProps` of the `DiagnosticParameter` with the role `DiagnosticRequestRoutineResults.response`. • `baseTypeEncoding` = BOOLEAN • `baseTypeSize` = 8 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
DOUBLE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is double (64 bit float).	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`	
Mapping Rule	**Mapping Type**
• `baseTypeEncoding` = IEEE754 • `baseTypeSize` = 64 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist Derivation from `fixedLength` =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
DOUBLE_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is double (64 bit float) array.	
Template Description	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = IEEE754 • `baseTypeSize` = 64 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01001) • `arraySizeSemantics` either does not exist or exists and is set to `fixedSize` (cf. TPS_DEXT_01001) Derivation from `fixedLength` =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
FLOAT	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is float.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

▽

△

Mapping Rule	Mapping Type
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
FLOAT_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is float array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16	ECUC-ENUMERATION-LITERAL-DEF

▽

△

BSW Description	
Type of the signal is sint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize	

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT32_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint32 array.	
Template Description	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C* baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT64	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint64.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**

▽

△

• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT64_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint64 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = 2C • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT8	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Type of the signal is sint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT8_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics, DataPrototype.swDataDefProps	

▽

△

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C* baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Type of the signal is uint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE, UTF-32 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT32_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT64	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Type of the data is uint64.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT64_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint64 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT8	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE,WINDOWS-1252,UTF-8 • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

▽

△

BSW Parameter	BSW Type
UINT8_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
rreferenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
VARIABLE_LENGTH	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint8[DcmDspRoutineParameterSize]. This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01002) • arraySizeSemantics exists and is set to ArraySizeSemanticsEnum.variableSize (cf. TPS_DEXT_01002)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool
BSW Parameter	**BSW Type**
DcmDspStopRoutineOutSignal	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Provide description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataOutN elements in the XXX_Stop function call.	
Template Description	
This represents the related dataElement of the DiagnosticParameter	
M2 Parameter	
DiagnosticAbstractParameter.dataElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00842]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType
BSW Parameter	**BSW Type**
DcmDspTextTableMapping	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.	
Template Description	

▽

△

This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.	
M2 Parameter	
CompuMethod	
Mapping Rule	**Mapping Type**
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00999]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspDiagnosisRepresentationDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The data value in the diagnosis representation.	
Template Description	
This represents a textual constant in the computation method.	
M2 Parameter	
CompuConstTextContent.vt	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01001]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspInternalDataValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
The ECU internal data value.	
Template Description	
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit:** This specifies the upper limit of a of the scale.	
M2 Parameter	
CompuScale.lowerLimit, CompuScale.upperLimit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01000]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal

BSW Parameter	BSW Type
DcmDspRoutineParameterSize	ECUC-INTEGER-PARAM-DEF

BSW Description
Provide the size of a RoutineControl parameter in bytes

Template Description
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter
DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
"Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00844]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal

BSW Parameter	BSW Type
DcmDspRoutineSignalEndianness	ECUC-ENUMERATION-PARAM-DEF

BSW Description
Defines the endianness of the data.

Template Description
This attribute specifies the byte order of the base type.

M2 Parameter
BaseTypeDirectDefinition.byteOrder

Mapping Rule	Mapping Type
• If the parent container is DcmDspRequestRoutineResultsOutSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response • If the parent container is DcmDspStopRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.request • If the parent container is DcmDspStopRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStop Routine.response • If the parent container is DcmDspStartRoutineInSignal: baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.request • If the parent container is DcmDspStartRoutineOutSignal: baseType.baseTypeDefinition.byte Order referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStart Routine.response • If the parent container is DcmDspRequestRoutineResultsInSignal: baseType.baseType Definition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01013]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal
BSW Parameter	**BSW Type**
DcmDspRoutineSignalPos	ECUC-INTEGER-PARAM-DEF
BSW Description	
Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.	
Template Description	
This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticDataIdentifier, DiagnosticParameterIdentifier, or DiagnosticRoutineSubfunction.	
M2 Parameter	
DiagnosticAbstractParameter.bitOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00843]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
BOOLEAN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is boolean.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = BOOLEAN • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
DOUBLE	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is double (64 bit float).

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = IEEE754 • `baseTypeSize` = 64 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist Derivation from `fixedLength` =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
DOUBLE_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is double (64 bit float) array.

Template Description

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = IEEE754 • `baseTypeSize` = 64 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01001) • `arraySizeSemantics` either does not exist or exists and is set to `fixedSize` (cf. TPS_DEXT_01001) Derivation from `fixedLength` =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
FLOAT	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is float.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
FLOAT_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is float array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT16	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is sint16.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DataPrototype.swDataDefProps`

Mapping Rule	Mapping Type
referenced by `swDataDefProps` of the `DiagnosticParameter` with the role `DiagnosticRequestRoutineResults.response`. • `baseTypeEncoding` = 2C • `baseTypeSize` = 16 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT16_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is sint16 array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

M2 Parameter

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT32_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint32 array.	
Template Description	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C* baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT64	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint64.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**

▽

△

• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
SINT64_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint64 array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	**Mapping Type**
• baseTypeEncoding = 2C • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**

▽

△

SINT8	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT8_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is sint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = 2C* baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
UINT16_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is uint16 array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticDataElement.arraySizeSemantics`, `DiagnosticDataElement.maxNumberOfElements`, `DataPrototype.swDataDefProps`

Mapping Rule	Mapping Type
referenced by `swDataDefProps` of the `DiagnosticParameter` with the role `DiagnosticRequestRoutineResults.response`. • `baseTypeEncoding` = NONE • `baseTypeSize` = 16 • `maxNumberOfElements` exists and value is greater than 0 (cf. TPS_DEXT_01001) • `arraySizeSemantics` either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
UINT32	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the signal is uint32.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DataPrototype.swDataDefProps:
This property allows to specify data definition properties which apply on data prototype level.

M2 Parameter

▽

△

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE, UTF-32 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT32_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
UINT64	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is uint64.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticValueNeeds.fixedLength:
This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency.

This attribute controls whether the data length of the data is fixed.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

`BaseTypeDirectDefinition.baseTypeEncoding`, `BaseTypeDirectDefinition.baseTypeSize`, `DiagnosticValueNeeds.fixedLength`, `DiagnosticDataElement.maxNumberOfElements`, `DiagnosticDataElement.arraySizeSemantics`

Mapping Rule	Mapping Type
• `baseTypeEncoding` = NONE • `baseTypeSize` = 64 • `maxNumberOfElements` shall not exist • `arraySizeSemantics` shall not exist Derivation from `fixedLength` =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
UINT64_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is uint64 array.

Template Description

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength:** This attribute is applicable only if the DiagnosticValueNeeds is aggregated within a BswModuleDependency. This attribute controls whether the data length of the data is fixed. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticValueNeeds.fixedLength, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics

Mapping Rule	Mapping Type
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to fixedSize (cf. TPS_DEXT_01001) Derivation from fixedLength =1 possible.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
UINT8	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DataPrototype.swDataDefProps, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**

▽

△

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE,WINDOWS-1252,UTF-8 • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT8_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DataPrototype.swDataDefProps:** This property allows to specify data definition properties which apply on data prototype level.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, DataPrototype.swDataDefProps	

Mapping Rule	**Mapping Type**
rreferenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to ArraySizeSemantics Enum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignalCompositePool/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**

▽

△

VARIABLE_LENGTH	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the signal is uint8[DcmDspRoutineParameterSize]. This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response. • baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01002) • arraySizeSemantics exists and is set to ArraySizeSemanticsEnum.variableSize (cf. TPS_DEXT_01002)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine
BSW Parameter	**BSW Type**
DcmDspStopRoutineRoleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to DcmDspAuthenticationRow that defines a role in that this routine can be stopped.	
Template Description	
This attribute allows for the specification of the position of the enclosing role in a bitfield of roles.	
M2 Parameter	
DiagnosticAuthRole.bitPosition	
Mapping Rule	**Mapping Type**
Applicable if the current access permission is referenced by DiagnosticStopRoutine. The value of DcmDspStopRoutineRole is calculated based on the bits of all referenced DiagnosticAuthRole.bit Position, where a value of 2^bitPosition is used for each DiagnosticAuthRole.bitPosition. The accumulated value represents a bitfield that has a bit set for each DiagnosticAuthRole.bitPosition.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01145]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**
DcmDspSecurityADRSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Size in bytes of the AccessDataRecord used in GetSeed	
Template Description	
This represents the size of the AccessDataRecord used in GetSeed. Unit:byte.	
M2 Parameter	
DiagnosticSecurityLevel.accessDataRecordSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00725]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**
DcmDspSecurityAttemptCounterEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This configuration parameter controls the existence of the APIs to set / get the attempt counter values towards application (Xxx_SetSecurityAttemptCounter() / Xxx_GetSecurityAttemptCounter()). In case of enabled, the security attempt counter values are passed to application, whenever there is a change in the value. This allows storing the values in nonvolatile RAM and restoring them at ECU startup.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01050]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**
DcmDspSecurityCompareKeyFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Function name to request the result of a key comparison. Parameter is only relevant if DcmDspSecurityUsePort=="USE_ ASYNCH_FNC". This parameter is related to the interface Xxx_CompareKey.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	

▽

△

DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00969]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**
DcmDspSecurityDelayTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Delay time after failed security access in seconds. This is started after DcmDspSecurityNumAttDelay number of failed security accesses. min: A negative value is not allowed.	
Template Description	
This represents the delay time after a failed security access. Unit: second.	
M2 Parameter	
DiagnosticSecurityLevel.securityDelayTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00757]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**
DcmDspSecurityDelayTimeOnBoot	ECUC-FLOAT-PARAM-DEF
BSW Description	
Value of the delay timer in case of 'power on' in seconds. This delay indicates the time at ECU boot power-on time during which the Dcm does not accept a security access. min: A negative value is not allowed.	
Template Description	
Start delay timer on power on in seconds. This delay indicates the time after ECU boot power-on where no security access request is accepted.	
M2 Parameter	
DiagnosticSecurityAccess.securityDelayTimeOnBoot	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00726]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**

▽

△

DcmDspSecurityGetAttemptCounterFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Function name to request the value of an attempt counter. This parameter is related to the interface Xxx_ GetSecurityAttempt Counter.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01048]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**
DcmDspSecurityGetSeedFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Callout function name used to request a seed. This parameter is related to the interface Xxx_GetSeed.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00968]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**
DcmDspSecurityKeySize	ECUC-INTEGER-PARAM-DEF
BSW Description	
size of the security key (in Bytes).	
Template Description	
This represents the size of the security key. Unit: byte.	
M2 Parameter	
DiagnosticSecurityLevel.keySize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00760]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**
DcmDspSecurityLevel	ECUC-INTEGER-PARAM-DEF
BSW Description	
Value of Security level. The locked state cannot be configured explicitly. 1,2,3...63: configuration dependent - Conversion formula to calculate SecurityLevel out of tester requested SecurityAccessType parameter: SecurityLevel = (SecurityAccessType (requestSeed) + 1) / 2 Type: Dcm_SecLevelType	
Template Description	
This would be 0x01, 0x03, 0x05, ... The sendKey id can be computed by adding 1 to the requestSeedId	
M2 Parameter	
DiagnosticSecurityAccess.requestSeedId	
Mapping Rule	**Mapping Type**
DcmDspSecurityLevel=(requestSeedId+1)/2	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00754]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**
DcmDspSecurityNumAttDelay	ECUC-INTEGER-PARAM-DEF
BSW Description	
Number of failed security accesses after which the delay time is activated	
Template Description	
This represents the number of failed security accesses after which the delay time is activated.	
M2 Parameter	
DiagnosticSecurityLevel.numFailedSecurityAccess	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00762]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**
DcmDspSecuritySeedSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
size of the security seed (in Bytes).	
Template Description	
This represents the size of the security seed. Unit: byte.	
M2 Parameter	
DiagnosticSecurityLevel.seedSize	



△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00755]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**
DcmDspSecuritySetAttemptCounterFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Function name to set the value of an attempt counter. This parameter is related to the interface Xxx_ SetSecurityAttempt Counter.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01049]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow/DcmDspSecurityUsePort
BSW Parameter	**BSW Type**
USE_ASYNCH_CLIENT_SERVER	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The DCM will access the data using an R-Port requiring a asynchronous ClientServerInterface SecurityAccess_{Security Level}. The R-Port is described in DcmDspSecurityRow description.	
Template Description	
This represents the ability to point into the component hierarchy (under possible consideration of the rootSoftware Composition)	
M2 Parameter	
DiagnosticServiceSwMapping.mappedSwcServiceDependencyInSystem	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow/DcmDspSecurityUsePort
BSW Parameter	**BSW Type**
USE_ASYNCH_FNC	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

The DCM will access the data using the functions that are defined in the parameters DcmDspSecurityGetSeedFnc and DcmDspSecurityCompareKeyFnc as well as the functions defined in DcmDspSecurityGetAttemptCounterFnc and DcmDspSecuritySetAttemptCounterFnc, if enabled by the parameter DcmDspSecurityAttemptCounterEnabled. DCM_E_PENDING return is allowed and OpStatus is existing as IN parameter.	
Template Description	
This represents the ability to point into the component hierarchy (under possible consideration of the rootSoftware Composition)	
M2 Parameter	
DiagnosticServiceSwMapping.mappedSwcServiceDependencyInSystem	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow/DcmDspSessionForBoot
BSW Parameter	**BSW Type**
DCM_NO_BOOT	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
This diagnostic session doesn't allow to jump to Bootloader.	
Template Description	
This diagnostic session doesn't allow to jump to Bootloader.	
M2 Parameter	
DiagnosticJumpToBootLoaderEnum.noBoot	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow/DcmDspSessionForBoot
BSW Parameter	**BSW Type**
DCM_OEM_BOOT	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
This diagnostic session allows to jump to OEM Bootloader and bootloader sends final response.	
Template Description	
This diagnostic session allows to jump to OEM Bootloader. In this case the bootloader send the final response.	
M2 Parameter	
DiagnosticJumpToBootLoaderEnum.oemBoot	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow/DcmDspSessionForBoot	
BSW Parameter	**BSW Type**	
DCM_OEM_BOOT_RESPAPP	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
This diagnostic session allows to jump to OEM Bootloader and application sends final response.		
Template Description		
This diagnostic session allows to jump to OEM Bootloader and application sends final response.		
M2 Parameter		
DiagnosticJumpToBootLoaderEnum.oemBootRespApp		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow/DcmDspSessionForBoot	
BSW Parameter	**BSW Type**	
DCM_SYS_BOOT	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
This diagnostic session allows to jump to System Supplier Bootloader and bootloader sends final response.		
Template Description		
This diagnostic session allows to jump to System Supplier Bootloader. In this case the bootloader send the final response.		
M2 Parameter		
DiagnosticJumpToBootLoaderEnum.systemSupplierBoot		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow/DcmDspSessionForBoot	
BSW Parameter	**BSW Type**	
DCM_SYS_BOOT_RESPAPP	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
This diagnostic session allows to jump to System Supplier Bootloader and application sends final response.		
Template Description		
This diagnostic session allows to jump to System Supplier Bootloader and application sends final response.		
M2 Parameter		
DiagnosticJumpToBootLoaderEnum.systemSupplierBootRespApp		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow
BSW Parameter	**BSW Type**
DcmDspSessionLevel	ECUC-INTEGER-PARAM-DEF
BSW Description	
subFunction value of the DiagnosticSession. 0, 127 and all values above 127 are reserved by ISO	
Template Description	
This is the numerical identifier used to identify the DiagnosticSession in the scope of diagnostic workflow	
M2 Parameter	
DiagnosticSession.id	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00765]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow
BSW Parameter	**BSW Type**
DcmDspSessionP2ServerMax	ECUC-FLOAT-PARAM-DEF
BSW Description	
This is the session value for P2ServerMax in seconds (per Session). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds. DCM configuration tools must convert this float value to the appropriate value format for the use in the software implementation of DCM. This value is reported to the tester within the response to the 'Session Control' service.	
Template Description	
This is the session value for P2ServerMax in seconds (per Session Control). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds.	
M2 Parameter	
DiagnosticSession.p2ServerMax	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00766]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow
BSW Parameter	**BSW Type**
DcmDspSessionP2StarServerMax	ECUC-FLOAT-PARAM-DEF
BSW Description	
This is the session value for P2*ServerMax in seconds (per Session). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds. DCM configuration tools must convert this float value to the appropriate value format for the use in the software implementation of DCM. This value is reported to the tester within the response to the 'Session Control' service.	
Template Description	

▽

△

This is the session value for P2*ServerMax in seconds (per Session Control). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds.	
M2 Parameter	
DiagnosticSession.p2StarServerMax	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00768]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp
BSW Parameter	**BSW Type**
DcmDspVehInfo	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration (parameters) for one single VehicleInfoType of service $09	
Template Description	
This meta-class represents the ability to model an instance of the OBD mode 0x09 service.	
M2 Parameter	
DiagnosticRequestVehicleInfo	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00630]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo
BSW Parameter	**BSW Type**
DcmDspVehInfoData	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Data Item of an InfoType; ShortName is post-fix of the port interface name.	
Template Description	
This meta-class represents the ability to describe a concrete piece of data to be taken into account for diagnostic purposes.	
M2 Parameter	
DiagnosticDataElement	
Mapping Rule	**Mapping Type**
The value shall be taken from DiagnosticRequestVehicleInfo.infoType.dataElement.dataElement.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00888]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData
BSW Parameter	**BSW Type**
DcmDspVehInfoDataOrder	ECUC-INTEGER-PARAM-DEF

▽

△

BSW Description	
Defines the order of the data item in the InfoType; values: 0..255; first data item having the order number 0; the next 1 and so on. The configuration of order needs to be unique per InfoType.	
Template Description	
This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticDataIdentifier, DiagnosticParameterIdentifier, or DiagnosticRoutineSubfunction.	
M2 Parameter	
DiagnosticParameter.bitOffset	
Mapping Rule	**Mapping Type**
Value shall be taken from DiagnosticRequestVehicleInfo.infoType.dataElement.bitOffset.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00891]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData
BSW Parameter	**BSW Type**
DcmDspVehInfoDataReadFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Callout function name for reading InfoType data item. Only required in case parameter 'DcmDspVehInfoDataUsePort' is set to 'false'	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
The BswServiceDependency should have aRoleBasedBswModuleEntryAssignment that in turn has attribute role set to Xxx_GetInfotypeValueData and points to a BswModuleEntry.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00889]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData
BSW Parameter	**BSW Type**
DcmDspVehInfoDataSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Size in bytes of the InfoType data item.	
Template Description	
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
Value shall be taken from DiagnosticRequestVehicleInfo.infoType.dataElement.dataElement.maxNumberOfElements.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00890]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData

BSW Parameter	BSW Type
DcmDspVehInfoDataUsePort	ECUC-BOOLEAN-PARAM-DEF

BSW Description

When this parameter is set to true the DCM will access the Data using an R-Port requiring a PortInterface IInfotype Services_{VehInfoData}. The R-Port is named InfotypeServices_{VehInfoData} where {VEHINFODATA} is the name of the container DcmDspVehInfoData. In that case, the DcmDspVehInfoDataReadFnc is ignored and the RTE APIs are used.

When this parameter is set to false, the DCM calls the function defined in DcmDspVehInfoDataReadFnc.

Template Description

DiagnosticServiceSwMapping.mappedBswServiceDependency:
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.

DiagnosticServiceSwMapping.mappedSwcServiceDependencyInSystem:
This represents the ability to point into the component hierarchy (under possible consideration of the rootSoftware Composition)

M2 Parameter

`DiagnosticServiceSwMapping.mappedBswServiceDependency`, `DiagnosticServiceSwMapping.mappedSwcServiceDependencyInSystem`

Mapping Rule	Mapping Type
Shall be set to TRUE if the reference DiagnosticServiceSwMapping.mappedSwcService Dependency exists. Shall be set to FALSE if the reference DiagnosticServiceSwMapping.mapped BswServiceDependency exists.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00727]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo

BSW Parameter	BSW Type
DcmDspVehInfoInfoType	ECUC-INTEGER-PARAM-DEF

BSW Description

value of InfoType.

Within each DcmConfigSet all DcmDspVehInfoInfoType values shall be unique.

Template Description

This attribute represents the value of InfoType (see SAE J1979-DA).

M2 Parameter

`DiagnosticInfoType.id`

Mapping Rule	Mapping Type
If DiagnosticRequestVehicleInfo, us DiagnosticRequestVehicleInfo.infoType.id.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00631]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions

BSW Parameter	BSW Type
DcmModeCondition	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

▽

△

This container contains the configuration of a mode condition or an environmental conditions which can be used as argument in DcmModeRules. One DcmModeCondition shall contain either one DcmSwcModeRef or one DcmBswModeRef or one DcmSwcSRData ElementRef. Please note that the Dcm acts as well as mode manager. Therefore the references DcmSwcModeRef or one DcmBswMode Ref. might point to provided ModeDeclarationGroupPrototypes of the Dcm itself as well as to provided ModeDeclaration GroupPrototypes of other Bsw Modules or software components. In case of a configured DcmSwcModeRef or DcmBswModeRef only the DcmConditionType DCM_EQUALS or DCM_ EQUALS_NOT are applicable. In case of DcmSwcSRDataElementRef all literals of DcmConditionType are possible.
Template Description
DiagnosticCompareConditions are atomic conditions. They are based on the idea of a comparison at runtime of some variable data with something constant. The type of the comparison (==, !=, <, <=, ...) is specified in DiagnosticCompare Condition.compareType.
M2 Parameter
DiagnosticEnvCompareCondition

Mapping Rule	**Mapping Type**
Depending on the reference a DcmModeCondition is mapped to a DiagnosticEnvModeCondition if only one reference is present and reference is a DcmSwcModeRef or a DcmBswModeRef. If two references are present, a DcmSwcSRDataElementRef and a DcmSwcSRDataElementValueRef, then DcmModeCondition is mapped to a DiagnosticEnvDataCondition.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00928]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition

BSW Parameter	**BSW Type**
DcmBswModeRef	ECUC-INSTANCE-REFERENCE-DEF
BSW Description	
This parameter references a mode of a ModeDeclarationGroupPrototype provided by a Basic Software Module used for the condition. Please note that such ModeDeclarationGroupPrototype are owned by a Basic Software Module Description in the role providedModeGroup.	
Template Description	
This reference represents both the ModeDeclarationGroupPrototype and the ModeDeclaration relevant for the mode comparison.	
M2 Parameter	
DiagnosticEnvModeCondition.modeElement	

Mapping Rule	**Mapping Type**
For DcmModeRef a new DiagnosticEnvBswModeElement is used, pointing to the ModeDeclaration via ModeInModuleDescriptionInstanceRef. This new DiagnosticEnvModeElement shall be aggregated by the same DiagnosticEnvironmentalConfition as the DiagnosticEnvModeCondition, in which the target of the reference DiagnosticEnvModeCondition.modeElement points to the this DiagnosticEnvModeElement.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00931]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType
BSW Parameter	**BSW Type**
DCM_EQUALS	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Equal (==)	
Template Description	
DiagnosticCompareTypeEnum.isEqual: equal **DiagnosticEnvCompareCondition.compareType:** This attributes represents the concrete type of the comparison.	
M2 Parameter	
DiagnosticCompareTypeEnum.isEqual, DiagnosticEnvCompareCondition.compareType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType
BSW Parameter	**BSW Type**
DCM_EQUALS_NOT	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Not equal (!=)	
Template Description	
DiagnosticCompareTypeEnum.isNotEqual: not equal **DiagnosticEnvCompareCondition.compareType:** This attributes represents the concrete type of the comparison.	
M2 Parameter	
DiagnosticCompareTypeEnum.isNotEqual, DiagnosticEnvCompareCondition.compareType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType
BSW Parameter	**BSW Type**
DCM_GREATER_OR_EQUAL	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Greater or equal (>=)	
Template Description	

▽

△

DiagnosticCompareTypeEnum.isGreaterOrEqual: greater than or equal **DiagnosticEnvCompareCondition.compareType:** This attributes represents the concrete type of the comparison.	
M2 Parameter	
DiagnosticCompareTypeEnum.isGreaterOrEqual, DiagnosticEnvCompareCondition.compareType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType
BSW Parameter	**BSW Type**
DCM_GREATER_THAN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Greater than (>)	
Template Description	
DiagnosticCompareTypeEnum.isGreaterThan: greater than **DiagnosticEnvCompareCondition.compareType:** This attributes represents the concrete type of the comparison.	
M2 Parameter	
DiagnosticCompareTypeEnum.isGreaterThan, DiagnosticEnvCompareCondition.compareType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType
BSW Parameter	**BSW Type**
DCM_LESS_OR_EQUAL	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Less or equal (<=)	
Template Description	
DiagnosticCompareTypeEnum.isLessOrEqual: less than or equal **DiagnosticEnvCompareCondition.compareType:** This attributes represents the concrete type of the comparison.	
M2 Parameter	
DiagnosticCompareTypeEnum.isLessOrEqual, DiagnosticEnvCompareCondition.compareType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType
BSW Parameter	**BSW Type**
DCM_LESS_THAN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Less than (<)	
Template Description	
DiagnosticCompareTypeEnum.isLessThan: less than **DiagnosticEnvCompareCondition.compareType:** This attributes represents the concrete type of the comparison.	
M2 Parameter	
DiagnosticCompareTypeEnum.isLessThan, DiagnosticEnvCompareCondition.compareType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmSwcDataElementValue/ DcmSwcDataElementArray/DcmSwcDataElementArrayElement
BSW Parameter	**BSW Type**
DcmSwcDataElementArrayElementIndex	ECUC-INTEGER-PARAM-DEF
BSW Description	
Index to an element of the compare value array.	
Template Description	
This attribute represents a fixed compare value taken to evaluate the compare condition.	
M2 Parameter	
DiagnosticEnvDataCondition.compareValue	
Mapping Rule	**Mapping Type**
DiagnosticDataElement referenced by the DiagnosticEnvDataCondition is an array.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01127]

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmSwcDataElementValue/ DcmSwcDataElementArray/DcmSwcDataElementArrayElement
BSW Parameter	**BSW Type**
DcmSwcDataElementArrayElementValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Value of an array element compare value.	
Template Description	
This attribute represents a fixed compare value taken to evaluate the compare condition.	
M2 Parameter	
DiagnosticEnvDataCondition.compareValue	
Mapping Rule	**Mapping Type**

▽

△

DiagnosticDataElement referenced by the DiagnosticEnvDataCondition is an array.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01128]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmSwcDataElementValue/ DcmSwcDataElementPrimitive
BSW Parameter	**BSW Type**
DcmSwcDataElementPrimitiveValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Primitive compare value.	
Template Description	
This attribute represents a fixed compare value taken to evaluate the compare condition.	
M2 Parameter	
DiagnosticEnvDataCondition.compareValue	
Mapping Rule	**Mapping Type**
DiagnosticDataElement referenced by the DiagnosticEnvDataCondition is primitive.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01126]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition
BSW Parameter	**BSW Type**
DcmSwcModeRef	ECUC-INSTANCE-REFERENCE-DEF
BSW Description	
This parameter references a mode in a particular mode request port of a software component that is used for the condition.	
Template Description	
This reference represents both the ModeDeclarationGroupPrototype and the ModeDeclaration relevant for the mode comparison.	
M2 Parameter	
DiagnosticEnvModeCondition.modeElement	
Mapping Rule	**Mapping Type**
For DcmModeRef a new DiagnosticEnvSwcModeElement is used, pointing to the ModeDeclaration via PModeInSystemInstanceRef. This new DiagnosticEnvModeElement shall be aggregated by the same DiagnosticEnvironmentalConfition as the DiagnosticEnvModeCondition, in which the target of the reference DiagnosticEnvModeCondition.modeElement points to the this DiagnosticEnvMode Element.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00930]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition
BSW Parameter	**BSW Type**
DcmSwcSRDataElementRef	ECUC-CHOICE-REFERENCE-DEF
BSW Description	

▽

△

Reference to environmental conditions. It is possible to reference a S/R Receiver-Port to read physical values and compare (equal, greater, less,...) them with a configured value that is defined by DcmSwcDataElementValue.	
Template Description	
This reference represents the related diagnostic data element.	
M2 Parameter	
DiagnosticEnvDataCondition.dataElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01037]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions
BSW Parameter	**BSW Type**
DcmModeRule	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration of a mode rule which represents a logical expression with DcmModeConditions or other DcmModeRules as arguments. All arguments are processed with the operator defined by DcmLogicalOperator, for instance: Argument_A AND Argument_B AND Argument_C	
Template Description	
A DiagnosticEnvConditionFormula embodies the computation instruction that is to be evaluated at runtime to determine if the DiagnosticEnvironmentalCondition is currently present (i.e. the formula is evaluated to true) or not (otherwise). The formula itself consists of parts which are combined by the logical operations specified by DiagnosticEnvConditionFormula.op. If a diagnostic functionality cannot be executed because an environmental condition fails then the diagnostic stack shall send a negative response code (NRC) back to the client. The value of the NRC is directly related to the specific formula and is therefore formalized in the attribute DiagnosticEnvConditionFormula.nrcValue.	
M2 Parameter	
DiagnosticEnvConditionFormula	
Mapping Rule	**Mapping Type**
A DcmModeRule is mapped to a DiagnosticEnvConditionFormula, if this DcmModeRule is a subrule, i.e. it is referenced by a DcmArgumentRef. In addition, a new DiagnosticEnvironmental Condition shall be created with DiagnosticEnvironmentalCondition.formula containing a Diagnostic EnvConditionFormula. In both cases, if no DcmLogicalOperator is present in this DcmModeRule, then DiagnosticEnvConditionFormula shall be set to DiagnosticLogicalOperatorEnum.logicalAnd.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00925]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeRule
BSW Parameter	**BSW Type**
DcmArgumentRef	ECUC-CHOICE-REFERENCE-DEF
BSW Description	
This is a choice reference either to a mode condition or a an other mode rule serving as sub-expression.	
Template Description	
A DiagnosticEnvConditionFormulaPart can either be a atomic condition, e.g. a DiagnosticEnvCompareCondition, or a DiagnosticEnvConditionFormula, again, which allows arbitrary nesting.	
M2 Parameter	

▽

△

DiagnosticEnvConditionFormulaPart	
Mapping Rule	**Mapping Type**
Depending on the destination, one DcmArgumentRef is mapped to a DiagnosticEnvCondition Formula if "destination" is a DcmModeRule, and to a DiagnosticEnvCompareCondition, if "destination" is a DcmModeCondition. The order of the aggregation of the DiagnosticEnvCondition FormulaParts shall correspond to the ordering of the index of the DcmArgumentRefs.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00927]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeRule/DcmLogicalOperator
BSW Parameter	**BSW Type**
DCM_AND	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
And (&&)	
Template Description	
DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticLogicalOperatorEnum.logicalAnd:** Logical AND	
M2 Parameter	
DiagnosticEnvConditionFormula.op, DiagnosticLogicalOperatorEnum.logicalAnd	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeRule/DcmLogicalOperator
BSW Parameter	**BSW Type**
DCM_OR	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Or (||)	
Template Description	
DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticLogicalOperatorEnum.logicalOr:** Logical OR	
M2 Parameter	
DiagnosticEnvConditionFormula.op, DiagnosticLogicalOperatorEnum.logicalOr	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeRule
BSW Parameter	**BSW Type**
DcmModeRuleNrcValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Optional parameter which defines the NRC to be sent in case the mode rule condition is not valid.	
Template Description	
This attribute represents the concrete NRC value that shall be returned if the condition fails.	
M2 Parameter	
DiagnosticEnvConditionFormula.nrcValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00949]

BSW Module	BSW Context
Dcm	Dcm/DcmGeneral
BSW Parameter	**BSW Type**
DcmDDDIDStorage	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This configuration switch defines, whether DDDID definition is stored non-volatile or not. true: DDDID are stored non-volatile false: DDDID are only maintained volatile	
Template Description	
This configuration switch defines whether DDDID definition is handled as non-volatile information or not.	
M2 Parameter	
DiagnosticDynamicallyDefineDataIdentifierClass.configurationHandling	
Mapping Rule	**Mapping Type**
volatile -> false, nonVolatile -> true	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00971]

BSW Module	BSW Context
Dcm	Dcm/DcmGeneral
BSW Parameter	**BSW Type**
DcmRespondAllRequest	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If set to FALSE the Dcm will not respond to diagnostic request that contains a service ID which is in the range from 0x40 to 0x7F or in the range from 0xC0 to 0xFF (Response IDs).	
Template Description	
If set to FALSE the DCM will not respond to diagnostic request that contains a service ID which is in the range from 0x40 to 0x7F or in the range from 0xC0 to 0xFF (Response IDs).	
M2 Parameter	
DiagnosticCommonProps.responseOnAllRequestSids	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_Dcm_00600]

BSW Module	BSW Context	
Dcm	Dcm/DcmGeneral	
BSW Parameter		**BSW Type**
DcmS3ServerTimeoutOverwrite		ECUC-FLOAT-PARAM-DEF
BSW Description		
Time for the server to keep a diagnostic session other than the defaultSession active while not receiving any diagnostic request message. This parameter overwrites the 5s recommend value of the S3server timeout in ISO 14229-2. The parameter value is defined in seconds.		
Template Description		
Time for the server to keep a diagnostic session other than the default session active while not receiving any diagnostic request message.		
M2 Parameter		
DiagnosticSessionControlClass.s3ServerTimeout		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dcm_01240]

BSW Module	BSW Context	
Dcm	Dcm/DcmGeneral	
BSW Parameter		**BSW Type**
DcmSecureCoding		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Contains references to diagnostic elements that are used for the secure coding process.		
Template Description		
This meta-class acts a mapping element to select diagnostic service instances that are used in the secure coding procedure. Besides, there are already defined diagnostic capabilities they furthermore have a additional semantics that is used during the secure coding process. In particular, this class references write data by identifier instances that mandatory require a secure coding and this class references a diagnostic routine control that is used to provide the signature and to persist the data in non-volatile memory.		
M2 Parameter		
DiagnosticSecureCodingMapping		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dcm_01245]

BSW Module	BSW Context	
Dcm	Dcm/DcmGeneral/DcmSecureCoding	
BSW Parameter		**BSW Type**
DcmSecureCodingDataIdentifier		ECUC-REFERENCE-DEF
BSW Description		

▽

△

Refers to a writeable data identifier that requires the secure coding process to write the data.	
Template Description	
Refer to write data by identifier instances that have the semantics to be a secure coding DID.	
M2 Parameter	
DiagnosticSecureCodingMapping.dataIdentifier	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01246]

BSW Module	**BSW Context**
Dcm	Dcm/DcmGeneral
BSW Parameter	**BSW Type**
DcmVinRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the Did containing the VIN Information. This parameter is needed for function Dcm_GetVin	
Template Description	
This meta-class represents the ability to model a diagnostic data identifier (DID) that is fully specified regarding the payload at configuration-time.	
M2 Parameter	
DiagnosticDataIdentifier	
Mapping Rule	**Mapping Type**
This reference shall only apply to a DiagnosticDataIdentifier where the attribute representsVin is set to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00984]

G.3 Dem

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDTC
BSW Parameter	**BSW Type**
DemDTCFunctionalUnit	ECUC-INTEGER-PARAM-DEF
BSW Description	
DTCFuncitonalUnit is a 1-byte value which identifies the corresponding basic vehicle / system function which reports the DTC. This parameter is necessary for the report of severity information. If this parameter is configured for no DTC, the Dem provides no DTC functional unit information.	
Template Description	
This attribute specifies a 1-byte value which identifies the corresponding basic vehicle / system function which reports the DTC. This parameter is necessary for the report of severity information.	
M2 Parameter	
DiagnosticTroubleCodeUds.functionalUnit	
Mapping Rule	**Mapping Type**

▽

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00643]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDTC
BSW Parameter	**BSW Type**
DemDTCSeverity	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
DTC severity according to ISO 14229-1. This parameter depends on the automotive manufacturer. If it is not configured, the value is counted as 'no severity'. If this parameter is configured for no DTC, the Dem provides no DTC severity information.	
Template Description	
DTC severity according to ISO 14229-1.	
M2 Parameter	
DiagnosticTroubleCodeUds.severity	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00645]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDTC/DemDTCSeverity
BSW Parameter	**BSW Type**
DEM_SEVERITY_CHECK_AT_NEXT_HALT	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Check at next halt	
Template Description	
Check at next halt.	
M2 Parameter	
DiagnosticUdsSeverityEnum.checkAtNextHalt	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDTC/DemDTCSeverity
BSW Parameter	**BSW Type**
DEM_SEVERITY_CHECK_IMMEDIATELY	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Check immediately	
Template Description	
Check immediately.	

▽

△

M2 Parameter	
DiagnosticUdsSeverityEnum.immediately	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTC/DemDTCSeverity
BSW Parameter	**BSW Type**
DEM_SEVERITY_MAINTENANCE_ONLY	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Maintenance required	
Template Description	
Maintenance required.	
M2 Parameter	
DiagnosticUdsSeverityEnum.maintenanceOnly	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTC/DemDTCSeverity
BSW Parameter	**BSW Type**
DEM_SEVERITY_NO_SEVERITY	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
No severity information available	
Template Description	
No severity information available.	
M2 Parameter	
DiagnosticUdsSeverityEnum.noSeverity	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTC
BSW Parameter	**BSW Type**
DemDtcOBDRelevance	ECUC-BOOLEAN-PARAM-DEF
BSW Description	

▽

△

Defines is this DTC is relevant for OBD. OBD relevant DTCs are reported in the OBD related APIs and it is possible to configure the calculation of further OBD related data such as readiness groups, IUMPR, etc. If set to false, the DTC is not OBD relevant If set to true, the DTC is OBD relevant.	
Template Description	
The attribute controls whether the enclosing DTC is relevant for OBD.	
M2 Parameter	
DiagnosticTroubleCodeProps.obdRelevance	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00994]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDTC
BSW Parameter	**BSW Type**
DemDtcValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
3 Byte value of the UDS Diagnostic Trouble Code (DTC). The value is unique within a fault memory. The ranges: 0x000000-0x0000FF and 0xFFFFFF are reserved by ISO 14229-1.	
Template Description	
Unique Diagnostic Trouble Code value for UDS.	
M2 Parameter	
DiagnosticTroubleCodeUds.udsDtcValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00887]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDTC
BSW Parameter	**BSW Type**
DemNvStorageStrategy	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines when the event related data of the DTC is stored in the NVRAM.	
Template Description	
Change description for Class immediateNvDataStorage in table "Table A.111: DiagnosticTroubleCodeProps": Switch to enable immediate storage triggering of an according event memory entry persistently to NVRAM. true: immediate non-volatile storage triggering on first occurrence and shutdown. false: immediate non-volatile storage triggering on shutdown.	
M2 Parameter	
DiagnosticTroubleCodeProps.immediateNvDataStorage	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00127]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTC/DemNvStorageStrategy
BSW Parameter	**BSW Type**
DURING_SHUTDOWN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Store during shutdown	
Template Description	
Change description for Class immediateNvDataStorage in table "Table A.111: DiagnosticTroubleCodeProps": Switch to enable immediate storage triggering of an according event memory entry persistently to NVRAM. true: immediate non-volatile storage triggering on first occurrence and shutdown. false: immediate non-volatile storage triggering on shutdown.	
M2 Parameter	
DiagnosticTroubleCodeProps.immediateNvDataStorage	
Mapping Rule	**Mapping Type**
immediateNvDataStorage == false	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTC/DemNvStorageStrategy
BSW Parameter	**BSW Type**
IMMEDIATE_AT_FIRST_OCCURRENCE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Store at first occurrence of DemEventMemoryEntryStorageTrigger trigger and during shutdown	
Template Description	
Change description for Class immediateNvDataStorage in table "Table A.111: DiagnosticTroubleCodeProps": Switch to enable immediate storage triggering of an according event memory entry persistently to NVRAM. true: immediate non-volatile storage triggering on first occurrence and shutdown. false: immediate non-volatile storage triggering on shutdown.	
M2 Parameter	
DiagnosticTroubleCodeProps.immediateNvDataStorage	
Mapping Rule	**Mapping Type**
immediateNvDataStorage == true	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTC
BSW Parameter	**BSW Type**
DemWWHOBDDTCClass	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
DTC Class according to ISO 14229-1 [2013 version]. This parameter depends on the automotive manufacturer. If it is not configured, the value is marked as 'unclassified'. If this parameter is configured for no DTC, the Dem provides no DTC WWHOBD class information.	
Template Description	

▽

△

This attribute is used to identify (if applicable) the corresponding severity class of an WWH-OBD DTC.	
M2 Parameter	
DiagnosticTroubleCodeUds.wwhObdDtcClass	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00912]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDTC/DemWWHOBDDTCClass
BSW Parameter	**BSW Type**
DEM_DTC_WWHOBD_CLASS_A	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Class A	
Template Description	
This attribute represents the severity class A.	
M2 Parameter	
DiagnosticWwhObdDtcClassEnum.demDtcWwhObdClassA	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDTC/DemWWHOBDDTCClass
BSW Parameter	**BSW Type**
DEM_DTC_WWHOBD_CLASS_B1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Class B1	
Template Description	
This attribute represents the severity class B1.	
M2 Parameter	
DiagnosticWwhObdDtcClassEnum.demDtcWwhObdClassB1	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDTC/DemWWHOBDDTCClass
BSW Parameter	**BSW Type**
DEM_DTC_WWHOBD_CLASS_B2	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Class B2	

▽

△

Template Description	
This attribute represents the severity class B2.	
M2 Parameter	
DiagnosticWwhObdDtcClassEnum.demDtcWwhObdClassB2	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTC/DemWWHOBDDTCClass
BSW Parameter	**BSW Type**
DEM_DTC_WWHOBD_CLASS_C	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Class C	
Template Description	
This attribute represents the severity class C.	
M2 Parameter	
DiagnosticWwhObdDtcClassEnum.demDtcWwhObdClassC	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTC/DemWWHOBDDTCClass
BSW Parameter	**BSW Type**
DEM_DTC_WWHOBD_CLASS_NOCLASS	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
No Class information	
Template Description	
This attribute represents the option to intentionally not describe a dedicated severity class of an WWH-OBD DTC.	
M2 Parameter	
DiagnosticWwhObdDtcClassEnum.demDtcWwhObdClassNoInformation	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTCAttributes
BSW Parameter	**BSW Type**
DemAgingCycleCounterThreshold	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

Number of aging cycles needed to unlearn/delete the event.	
Template Description	
Number of aging cycles needed to unlearn/delete the event.	
M2 Parameter	
DiagnosticAging.threshold	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00623]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDTCAttributes
BSW Parameter	**BSW Type**
DemAgingCycleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the cycle which is triggering the aging of the event.	
Template Description	
This represents the applicable aging cycle.	
M2 Parameter	
DiagnosticAging.agingCycle	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00624]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDTCAttributes
BSW Parameter	**BSW Type**
DemDTCPriority	ECUC-INTEGER-PARAM-DEF
BSW Description	
Priority of the event/dtc, in view of full event memory. A lower value means higher priority.	
Template Description	
Priority of the event, in view of full event buffer. A lower value means higher priority.	
M2 Parameter	
DiagnosticTroubleCodeProps.priority	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00662]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDTCAttributes
BSW Parameter	**BSW Type**

▽

△

DemDTCSignificance	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Significance of the event, which indicates additional information concerning fault classification and resolution. It can be mapped as Dem-internal data element. It shall be configured, if it is a part of event related data.	
Template Description	
Significance of the event, which indicates additional information concerning fault classification and resolution.	
M2 Parameter	
DiagnosticTroubleCodeProps.significance	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00779]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDTCAttributes/DemDTCSignificance
BSW Parameter	**BSW Type**
DEM_EVENT_SIGNIFICANCE_OCCURRENCE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
issue, which indicates additional information concerning insufficient system behavior	
Template Description	
Issue, which indicates additional information concerning insufficient system behavior.	
M2 Parameter	
DiagnosticSignificanceEnum.occurence	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDTCAttributes
BSW Parameter	**BSW Type**
DemMaxNumberFreezeFrameRecords	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the number of according freeze frame records, which can maximal be stored for this event. Therefore all these freeze frame records have the same freeze frame class. This parameter is only required for calculated record numeration (refer to DemTypeOfFreezeFrameRecordNumeration).	
Template Description	
This attribute defines the number of according freeze frame records, which can maximal be stored for this event. Therefore all these freeze frame records have the same freeze frame class.	
M2 Parameter	
DiagnosticTroubleCodeProps.maxNumberFreezeFrameRecords	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00605]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTCAttributes
BSW Parameter	**BSW Type**
DemMemoryDestinationRef	ECUC-CHOICE-REFERENCE-DEF
BSW Description	
The memory destination assigns DTCs to one memory destination.	
Template Description	
Reference to the applicable DiagnosticMemoryDestination.	
M2 Parameter	
DiagnosticTroubleCodeProps.diagnosticMemory	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00890]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass
BSW Parameter	**BSW Type**
DemCounterBasedFdcThresholdStorageValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Threshold to allocate an event memory entry and to capture the Freeze Frame.	
Template Description	
Threshold to allocate an event memory entry and to capture the Freeze Frame.	
M2 Parameter	
DiagEventDebounceCounterBased.counterBasedFdcThresholdStorageValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00914]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass
BSW Parameter	**BSW Type**
DemDebounceBehavior	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines how the event debounce algorithm will behave, if a related enable condition is not fulfilled or Control DTCSetting of the related event is disabled.	
Template Description	
This attribute defines how the event debounce algorithm will behave, if a related enable condition is not fulfilled or Control DTCSetting of the related event is disabled.	
M2 Parameter	
DiagnosticDebounceAlgorithmProps.debounceBehavior	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00786]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass/DemDebounceBehavior
BSW Parameter	**BSW Type**
DEM_DEBOUNCE_FREEZE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The event debounce counter/timer will be frozen with the current value and will not change while a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. After all related enable conditions are fulfilled and Control DTCSetting of the related event is enabled again, the event qualification will continue with the next report of the event (i.e. Set EventStatus).	
Template Description	
The event debounce counter will be frozen with the current value and will not change while a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. After all related enable conditions are fulfilled and Control DTCSetting of the related event is enabled again, the event qualification will continue with the next report of the event (i.e. Set EventStatus).	
M2 Parameter	
DiagnosticDebounceBehaviorEnum.freeze	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass/DemDebounceBehavior
BSW Parameter	**BSW Type**
DEM_DEBOUNCE_RESET	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The event debounce counter/timer will be reset to initial value if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. The qualification of the event will be restarted with the next valid event report.	
Template Description	
The event debounce counter will be reset to initial value if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. The qualification of the event will be restarted with the next valid event report.	
M2 Parameter	
DiagnosticDebounceBehaviorEnum.reset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass
BSW Parameter	**BSW Type**
DemDebounceCounterDecrementStepSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the step size for decrementation of the internal debounce counter (PREPASSED).	
Template Description	
This value shall be taken to decrement the internal debounce counter.	
M2 Parameter	
DiagEventDebounceCounterBased.counterDecrementStepSize	



△

Mapping Rule	Mapping Type
Shall be taken from DiagnosticExtract::DiagnosticCommonProps.debounceAlgorithm Props.debounceAlgorithm.counterDecrementStepSize. Applicable if DiagnosticExtract::DiagnosticCommonProps.debounceAlgorithmProps.debounce Algorithm is modeled by means of a DiagEventDebounceCounterBased.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00635]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass
BSW Parameter	**BSW Type**
DemDebounceCounterFailedThreshold	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the value of the internal debounce counter, which indicates the failed status.	
Template Description	
This value defines the event-specific limit that indicates the "failed" counter status.	
M2 Parameter	
DiagEventDebounceCounterBased.counterFailedThreshold	

Mapping Rule	Mapping Type
Shall be taken from DiagnosticExtract::DiagnosticCommonProps.debounceAlgorithm Props.debounceAlgorithm.counterFailedThreshold. Applicable if DiagnosticExtract::DiagnosticCommonProps.debounceAlgorithmProps.debounce Algorithm is modeled by means of a DiagEventDebounceCounterBased.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00618]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass
BSW Parameter	**BSW Type**
DemDebounceCounterIncrementStepSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the step size for incrementation of the internal debounce counter (PREFAILED).	
Template Description	
This value shall be taken to increment the internal debounce counter.	
M2 Parameter	
DiagEventDebounceCounterBased.counterIncrementStepSize	

Mapping Rule	Mapping Type
Shall be taken from DiagnosticExtract::DiagnosticCommonProps.debounceAlgorithm Props.debounceAlgorithm.counterIncrementStepSize. Applicable if DiagnosticExtract::DiagnosticCommonProps.debounceAlgorithmProps.debounce Algorithm is modeled by means of a DiagEventDebounceCounterBased.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00637]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass
BSW Parameter	**BSW Type**
DemDebounceCounterJumpDown	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Switch for the activation of Jump-Down. true: Jump-Down activated false: Jump-Down deactivated	
Template Description	
This value activates or deactivates the counter jump-down behavior.	
M2 Parameter	
DiagEventDebounceCounterBased.counterJumpDown	
Mapping Rule	**Mapping Type**
Shall be taken from DiagnosticExtract::DiagnosticCommonProps.debounceAlgorithm Props.debounceAlgorithm.counterJumpDown. Applicable if DiagnosticExtract::DiagnosticCommonProps.debounceAlgorithmProps.debounce Algorithm is modeled by means of a DiagEventDebounceCounterBased.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00685]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass
BSW Parameter	**BSW Type**
DemDebounceCounterJumpDownValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Jump-Down value of the internal debounce counter which is taken as initialization value for the counter when the respective step-down occurs.	
Template Description	
This value represents the initial value of the internal debounce counter if the counting direction changes from incrementing to decrementing.	
M2 Parameter	
DiagEventDebounceCounterBased.counterJumpDownValue	
Mapping Rule	**Mapping Type**
Shall be taken from DiagnosticExtract::DiagnosticCommonProps.debounceAlgorithm Props.debounceAlgorithm.counterJumpDownValue. Applicable if DiagnosticExtract::DiagnosticCommonProps.debounceAlgorithmProps.debounce Algorithm is modeled by means of a DiagEventDebounceCounterBased.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00638]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass
BSW Parameter	**BSW Type**
DemDebounceCounterJumpUp	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Switch for the activation of Jump-Up. true: Jump-Up activated false: Jump-Up deactivated	



△

Template Description	
This value activates or deactivates the counter jump-up behavior.	
M2 Parameter	
DiagEventDebounceCounterBased.counterJumpUp	
Mapping Rule	**Mapping Type**
Shall be taken from DiagnosticExtract::DiagnosticCommonProps.debounceAlgorithm Props.debounceAlgorithm.counterJumpUp. Applicable if DiagnosticExtract::DiagnosticCommonProps.debounceAlgorithmProps.debounce Algorithm is modeled by means of a DiagEventDebounceCounterBased.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00686]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass
BSW Parameter	**BSW Type**
DemDebounceCounterJumpUpValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Jump-Up value of the internal debounce counter which is taken as initialization value for the counter when the respective step-up occurs.	
Template Description	
This value represents the initial value of the internal debounce counter if the counting direction changes from decrementing to incrementing.	
M2 Parameter	
DiagEventDebounceCounterBased.counterJumpUpValue	
Mapping Rule	**Mapping Type**
Shall be taken from DiagnosticExtract::DiagnosticCommonProps.debounceAlgorithm Props.debounceAlgorithm.counterJumpValue. Applicable if DiagnosticExtract::DiagnosticCommonProps.debounceAlgorithmProps.debounce Algorithm is modeled by means of a DiagEventDebounceCounterBased.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00639]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass
BSW Parameter	**BSW Type**
DemDebounceCounterPassedThreshold	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the value of the internal debounce counter, which indicates the passed status.	
Template Description	
This value defines the event-specific limit that indicates the "passed" counter status.	
M2 Parameter	
DiagEventDebounceCounterBased.counterPassedThreshold	
Mapping Rule	**Mapping Type**

▽

△

Shall be taken from DiagnosticExtract::DiagnosticCommonProps.debounceAlgorithm Props.debounceAlgorithm.counterPassedThreshold. Applicable if DiagnosticExtract::DiagnosticCommonProps.debounceAlgorithmProps.debounce Algorithm is modeled by means of a DiagEventDebounceCounterBased.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00636]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass
BSW Parameter	**BSW Type**
DemDebounceCounterStorage	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Switch to store the debounce counter value non-volatile or not. true: debounce counter value shall be stored non-volatile false: debounce counter value is volatile	
Template Description	
Switch to store the debounce counter value non-volatile or not. true: debounce counter value shall be stored non-volatile false: debounce counter value is volatile Please note that this attribute is not relevant for the adaptive platform.	
M2 Parameter	
DiagnosticDebounceAlgorithmProps.debounceCounterStorage	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00791]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDebounceTimeBasedClass
BSW Parameter	**BSW Type**
DemDebounceBehavior	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines how the event debounce algorithm will behave, if a related enable condition is not fulfilled or Control DTCSetting of the related event is disabled.	
Template Description	
This attribute defines how the event debounce algorithm will behave, if a related enable condition is not fulfilled or Control DTCSetting of the related event is disabled.	
M2 Parameter	
DiagnosticDebounceAlgorithmProps.debounceBehavior	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00786]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDebounceTimeBasedClass/DemDebounceBehavior
BSW Parameter	**BSW Type**
DEM_DEBOUNCE_FREEZE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The event debounce counter/timer will be frozen with the current value and will not change while a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. After all related enable conditions are fulfilled and Control DTCSetting of the related event is enabled again, the event qualification will continue with the next report of the event (i.e. Set EventStatus).	
Template Description	
The event debounce counter will be frozen with the current value and will not change while a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. After all related enable conditions are fulfilled and Control DTCSetting of the related event is enabled again, the event qualification will continue with the next report of the event (i.e. Set EventStatus).	
M2 Parameter	
DiagnosticDebounceBehaviorEnum.freeze	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDebounceTimeBasedClass/DemDebounceBehavior
BSW Parameter	**BSW Type**
DEM_DEBOUNCE_RESET	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The event debounce counter/timer will be reset to initial value if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. The qualification of the event will be restarted with the next valid event report.	
Template Description	
The event debounce counter will be reset to initial value if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. The qualification of the event will be restarted with the next valid event report.	
M2 Parameter	
DiagnosticDebounceBehaviorEnum.reset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDebounceTimeBasedClass
BSW Parameter	**BSW Type**
DemDebounceTimeFailedThreshold	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the time out duration for "Event Failed" qualification. The AUTOSAR configuration standard is to use SI units, so this parameter is defined as float value in seconds. Dem configuration tools must convert this float value to the appropriate value format for the use in the software implementation of Dem.	
Template Description	



△

This value represents the event-specific delay indicating the "failed" status.	
M2 Parameter	
DiagEventDebounceTimeBased.timeFailedThreshold	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00716]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDebounceTimeBasedClass
BSW Parameter	**BSW Type**
DemDebounceTimePassedThreshold	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the time out duration for "Event Passed" qualification. The AUTOSAR configuration standard is to use SI units, so this parameter is defined as float value in seconds. Dem configuration tools must convert this float value to the appropriate value format for the use in the software implementation of Dem.	
Template Description	
This value represents the event-specific delay indicating the "passed" status.	
M2 Parameter	
DiagEventDebounceTimeBased.timePassedThreshold	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00717]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDebounceTimeBasedClass
BSW Parameter	**BSW Type**
DemTimeBasedFdcThresholdStorageValue	ECUC-FLOAT-PARAM-DEF
BSW Description	
Threshold to allocate an event memory entry and to capture the Freeze Frame.	
Template Description	
Threshold to allocate an event memory entry and to capture the Freeze Frame.	
M2 Parameter	
DiagEventDebounceTimeBased.timeBasedFdcThresholdStorageValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00915]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDtrs/DemDtr
BSW Parameter	**BSW Type**

▽

△

DemDtrEventRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the DemEventParameter this DTR is related to. If the related event is not configured, the Dem cannot ensure consistency between the DTR and the event.	
Template Description	
DiagnosticTestResult: This meta-class represents the ability to define diagnostic test results. **DiagnosticEvent:** This element is used to configure DiagnosticEvents.	
M2 Parameter	
DiagnosticTestResult, DiagnosticEvent	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00808]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDtrs/DemDtr
BSW Parameter	**BSW Type**
DemDtrMid	ECUC-INTEGER-PARAM-DEF
BSW Description	
The OBDMID of the DTR. The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, 0xE0 are reserved.	
Template Description	
This represents the numerical measurement Id	
M2 Parameter	
DiagnosticMeasurementIdentifier.obdMid	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00809]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDtrs/DemDtr
BSW Parameter	**BSW Type**
DemDtrTid	ECUC-INTEGER-PARAM-DEF
BSW Description	
The OBDTID of the DTR.	
Template Description	
This represents the numerical id associated with the diagnostic test identifier.	
M2 Parameter	
DiagnosticTestIdentifier.id	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00810]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDtrs/DemDtr
BSW Parameter	**BSW Type**
DemDtrUasid	ECUC-INTEGER-PARAM-DEF
BSW Description	
The UaSId the DTR data shall be scaled to, and reported together with the rescaled DTR data.	
Template Description	
This represents the unit and scaling Id of the diagnostic test result.	
M2 Parameter	
DiagnosticTestIdentifier.uasId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00811]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDtrs/DemDtr
BSW Parameter	**BSW Type**
DemDtrUpdateKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Update conditions applied by the Dem to reports of DTR values. Only supported if a related Event is configured	
Template Description	
This attribute controls the update behavior of the enclosing DiagnosticTestResult.	
M2 Parameter	
DiagnosticTestResult.updateKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00812]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDtrs/DemDtr/DemDtrUpdateKind
BSW Parameter	**BSW Type**
DEM_DTR_UPDATE_ALWAYS	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Any DTR result reported by the monitor is used by the Dem.	
Template Description	
Any DTR result reported by the monitor is used by the Dem.	
M2 Parameter	
DiagnosticTestResultUpdateEnum.always	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDtrs/DemDtr/DemDtrUpdateKind
BSW Parameter	**BSW Type**
DEM_DTR_UPDATE_STEADY	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The Dem accepts reported DTRs only when the configured debouncing mechanism is stable at the FAIL or PASS limit.	
Template Description	
The Dem accepts reported DTRs only when the configured debouncing mechanism is stable at the FAIL or PASS limit.	
M2 Parameter	
DiagnosticTestResultUpdateEnum.steady	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter
BSW Parameter	**BSW Type**
DemCallbackClearEventAllowed	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The presence of this container indicates that the Dem has access to a "ClearEventAllowed" callback. In case there is a DemCallbackClearEventAllowedFnc, this parameter defines the name of the function that the Dem will call. In case there is no DemCallbackClearEventAllowedFnc, the Dem will have an R-Port requiring the interface CallbackClear EventAllowed whose name is generated by using the unique callback-prefix followed by the event name.	
Template Description	
In case the clearing of a Diagnostic Event has to be allowed or prohibited through the SWC interface CallbackClearEvent Allowed, the SWC has to indicate this by defining appropriate ServiceNeeds (i.e. DiagnosticEventNeeds).	
M2 Parameter	
DiagnosticEventClearAllowedEnum.requiresCallbackExecution	
Mapping Rule	**Mapping Type**
Shall be taken from DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticEvent ClearAllowedEnum.requiresCallbackExecution. If literal requiresCallbackExecution is set, the BSW container DemCallbackClearEventAllowed will be present. Dem will call the funtion named by configuration parameter DemCallbackClearEvent AllowedFnc. If no such function name is provided, Dem will require an R-Port with the interface name using the unique callback-prefix followed by the event name.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00607]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemCallbackClearEventAllowed
BSW Parameter	**BSW Type**
DemCallbackClearEventAllowedFnc	ECUC-FUNCTION-NAME-DEF
BSW Description	
Function name of prototype "ClearEventAllowed".	
Template Description	
This attribute defines whether the Dem has access to a "ClearEventAllowed" callback.	

▽

△

M2 Parameter	
DiagnosticEvent.eventClearAllowed	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00609]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemCallbackClearEventAllowed
BSW Parameter	**BSW Type**
DemClearEventAllowedBehavior	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the resulting UDS status byte for the related event, which must not be cleared according to the ClearEventAllowed callback.	
Template Description	
This attribute defines the resulting UDS status byte for the related event, which shall not be cleared according to the Clear EventAllowed callback	
M2 Parameter	
DiagnosticEvent.clearEventAllowedBehavior	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00788]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemCallbackClearEventAllowed/DemClearEvent AllowedBehavior
BSW Parameter	**BSW Type**
DEM_NO_STATUS_BYTE_CHANGE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The UDS status byte keeps unchanged.	
Template Description	
The event status byte keeps unchanged.	
M2 Parameter	
DiagnosticClearEventAllowedBehaviorEnum.noStatusByteChange	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemCallbackClearEventAllowed/DemClearEvent AllowedBehavior
BSW Parameter	**BSW Type**
DEM_ONLY_THIS_CYCLE_AND_READINESS	ECUC-ENUMERATION-LITERAL-DEF

▽

△

BSW Description	
The <...>ThisOperationCycle and readiness bits of the UDS status byte are reset.	
Template Description	
The OperationCycle and readiness bits of the event status byte are reset.	
M2 Parameter	
DiagnosticClearEventAllowedBehaviorEnum.onlyThisCycleAndReadiness	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter
BSW Parameter	**BSW Type**
DemDebounceAlgorithmClass	ECUC-CHOICE-CONTAINER-DEF
BSW Description	
Debounce algorithm class: counter based, time based, or monitor internal.	
Template Description	
This class represents the ability to specify the pre-debounce algorithm which is selected and/or required by the particular monitor. This class inherits from Identifiable in order to allow further documentation of the expected or implemented debouncing and to use the category for the identification of the expected / implemented debouncing.	
M2 Parameter	
DiagEventDebounceAlgorithm	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00604]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass
BSW Parameter	**BSW Type**
DemDebounceCounterBased	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration (parameters) for counter based debouncing.	
Template Description	
This meta-class represents the ability to indicate that the counter-based debounce algorithm shall be used by the DEM for this diagnostic monitor. This is related to set the ECUC choice container DemDebounceAlgorithmClass to DemDebounceCounterBased.	
M2 Parameter	
DiagEventDebounceCounterBased	
Mapping Rule	**Mapping Type**

▽

△

There are two ways to derive the existence of DemDebounceCounterBased: 1. DiagEventNeeds,diagEventDebounceAlgoritm exists and is modeled as a DiagEventDebounce CounterBased. 2. DiagnosticContributionSet.commonProperties.debounceAlgorithmProps.debounceAlgorithm exists and is modeled as a DiagEventDebounceCounterBased If both alternatives exist at the same time then the definition ot DiagnosticContributionSet.common Properties.debounceAlgorithmProps.debounceAlgorithm shall be handled with priority.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00711]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass
BSW Parameter	**BSW Type**
DemDebounceMonitorInternal	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration (parameters) for monitor internal debouncing.	
Template Description	
This meta-class represents the ability to indicate that no Dem pre-debounce algorithm shall be used for this diagnostic monitor. The SWC might implement an internal debouncing algorithm and report qualified (debounced) results to the Dem/ DM.	
M2 Parameter	
DiagEventDebounceMonitorInternal	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00712]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass
BSW Parameter	**BSW Type**
DemDebounceTimeBased	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration (parameters) for time based debouncing.	
Template Description	
This meta-class represents the ability to indicate that the time-based pre-debounce algorithm shall be used by the Dem for this diagnostic monitor. This is related to set the EcuC choice container DemDebounceAlgorithmClass to DemDebounceTimeBase.	
M2 Parameter	
DiagEventDebounceTimeBased	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00713]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter
BSW Parameter	**BSW Type**
DemEnableConditionGroupRef	ECUC-REFERENCE-DEF
BSW Description	
References an enable condition group.	
Template Description	
Defines which EnableConditionGroup is applicable for a DiagnosticEvent.	
M2 Parameter	
DiagnosticEventToEnableConditionGroupMapping	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00746]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter
BSW Parameter	**BSW Type**
DemEventAssociatedIdentification	ECUC-INTEGER-PARAM-DEF
BSW Description	
Event associated identifier that allows to identify an event. This value can be reported as internal data element in snapshot records or extended data records.	
Template Description	
This attribute represents the identification number that is associated with the enclosing DiagnosticEvent and allows to identify it when placed into a snapshot record or extended data record storage. This value can be reported as internal data element in snapshot records or extended data records.	
M2 Parameter	
DiagnosticEvent.associatedEventIdentification	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00969]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter
BSW Parameter	**BSW Type**
DemEventConfirmationThreshold	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the operation cycle threshold of the DTC confirmation status according "Confirmation Threshold" of ISO 14229-1.	
Template Description	
This attribute defines the number of operation cycles with a failed result before a confirmed DTC is set to 1. The semantic of this attribute is a by "1" increased value compared to the confirmation threshold of the "trip counter" mentioned in ISO 14229-1 in figure D.4. A value of "1" defines the immediate confirmation of the DTC along with the first reported failed. This is also sometimes called "zero trip DTC". A value of "2" defines a DTC confirmation in the operation cycle after the first occurred failed. A value of "2" is typically used in the US for OBD DTC confirmation.	
M2 Parameter	
DiagnosticEvent.confirmationThreshold	



△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00924]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter
BSW Parameter	**BSW Type**
DemEventKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter is used to distinguish between SW-C and BSW events.	
Template Description	
This attribute is used to distinguish between SWC and BSW events.	
M2 Parameter	
DiagnosticEvent.eventKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00660]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemEventKind
BSW Parameter	**BSW Type**
DEM_EVENT_KIND_BSW	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The event is a assigned to a BSW module	
Template Description	
The event is assigned to a BSW module.	
M2 Parameter	
DiagnosticEventKindEnum.bsw	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemEventKind
BSW Parameter	**BSW Type**
DEM_EVENT_KIND_SWC	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The event is a assigned to a SW-C	
Template Description	
The event is assigned to a SWC.	
M2 Parameter	

▽

△

DiagnosticEventKindEnum.swc	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemEventParameter
BSW Parameter	**BSW Type**
DemEventRecoverableInSameOperationCycle	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If parameter is configured to FALSE, reporting of PASSED will be ignored if the event is already "testfailed this operation cycle".	
Template Description	
If the attribute is set to true then reporting PASSED will reset the indication of a failed test in the current operation cycle. If the attribute is set to false then reporting PASSED will be ignored and not lead to a reset of the indication of a failed test.	
M2 Parameter	
DiagnosticEvent.recoverableInSameOperationCycle	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00916]

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemEventParameter
BSW Parameter	**BSW Type**
DemFFPrestorageInNvm	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If the event uses a pre-stored freeze-frame this attribute indicates if the event requires the pre-stored data to be stored in non-volatile memory. TRUE = store the pre-stored data in non-volatile memory, FALSE = pre-stored data is not stored in non-volatile memory.	
Template Description	
DiagnosticEventNeeds.prestoredFreezeframeStoredInNvm: If the Event uses a prestored freeze-frame (using the operations PrestoreFreezeFrame and ClearPrestoredFreezeFrame of the service interface DiagnosticMonitor) this attribute indicates if the Event requires the data to be stored in non-volatile memory. TRUE = Dem shall store the prestored data in non-volatile memory, FALSE = Data can be lost at shutdown (not stored in Nvm). **DiagnosticEvent.prestoredFreezeframeStoredInNvm:** If the Event uses a prestored freeze-frame (using the operations PrestoreFreezeFrame and ClearPrestoredFreezeFrame of the service interface DiagnosticMonitor) this attribute indicates if the Event requires the data to be stored in non-volatile memory. TRUE = Dem shall store the prestored data in non-volatile memory, FALSE = Data can be lost at shutdown (not stored in Nvm)	
M2 Parameter	
DiagnosticEventNeeds.prestoredFreezeframeStoredInNvm, DiagnosticEvent.prestoredFreezeframeStoredInNvm	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00948]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter
BSW Parameter	**BSW Type**
DemFFPrestorageSupported	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If this parameter is set to true, then the Prestorage of FreezeFrames is supported by the assigned event. This parameter is useful to calculate the buffer size.	
Template Description	
This attribute describes whether the Prestorage of FreezeFrames is supported by the assigned event or not. true: Prestorage of FreezeFrames is supported fFalse: Prestorage of FreezeFrames is not supported	
M2 Parameter	
DiagnosticEvent.prestorageFreezeFrame	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00671]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemIndicatorAttribute
BSW Parameter	**BSW Type**
DemIndicatorBehaviour	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Behaviour of the linked indicator	
Template Description	
Behavior of the linked indicator.	
M2 Parameter	
DiagnosticConnectedIndicator.behavior	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00682]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemIndicatorAttribute/DemIndicatorBehaviour
BSW Parameter	**BSW Type**
DEM_INDICATOR_BLINKING	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The indicator blinks when the event has status FAILED Not relevant with J1939.	
Template Description	
The indicator blinks when the event has status FAILED.	
M2 Parameter	
DiagnosticConnectedIndicatorBehaviorEnum.blinkMode	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemEventParameter/DemIndicatorAttribute/DemIndicatorBehaviour	
BSW Parameter	**BSW Type**	
DEM_INDICATOR_BLINK_CONT	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
The indicator is active and blinks when the event has status FAILED Not relevant with J1939.		
Template Description		
The indicator is active and blinks when the event has status FAILED.		
M2 Parameter		
DiagnosticConnectedIndicatorBehaviorEnum.blinkOrContinuousOnMode		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemEventParameter/DemIndicatorAttribute/DemIndicatorBehaviour	
BSW Parameter	**BSW Type**	
DEM_INDICATOR_CONTINUOUS	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
The indicator is active when the even has status FAILED		
Template Description		
The indicator is active when the event has status FAILED.		
M2 Parameter		
DiagnosticConnectedIndicatorBehaviorEnum.continuousOnMode		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemEventParameter/DemIndicatorAttribute/DemIndicatorBehaviour	
BSW Parameter	**BSW Type**	
DEM_INDICATOR_FAST_FLASH	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
Flash Indicator Lamp should be set to 'Fast Flash'		
Template Description		
Flash Indicator Lamp should be set to "Fast Flash".		
M2 Parameter		
DiagnosticConnectedIndicatorBehaviorEnum.fastFlashingMode		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemIndicatorAttribute/DemIndicatorBehaviour
BSW Parameter	**BSW Type**
DEM_INDICATOR_SLOW_FLASH	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Flash Indicator Lamp should be set to 'Slow Flash'	
Template Description	
Flash Indicator Lamp should be set to "Slow Flash".	
M2 Parameter	
DiagnosticConnectedIndicatorBehaviorEnum.slowFlashingMode	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemIndicatorAttribute
BSW Parameter	**BSW Type**
DemIndicatorFailureCycleCounterThreshold	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the number of failure cycles for the WarningIndicatorOnCriteria.	
Template Description	
This attribute defines the number of failure cycles for the WarningIndicatorOnCriteria. Please note that this attribute is not relevant for the Adaptive Platform.	
M2 Parameter	
DiagnosticConnectedIndicator.indicatorFailureCycleCounterThreshold	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00750]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemIndicatorAttribute
BSW Parameter	**BSW Type**
DemIndicatorHealingCycleCounterThreshold	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the number of healing cycles for the WarningIndicatorOffCriteria.	
Template Description	
This attribute defines the number of healing cycles for the WarningIndicatorOffCriteria	
M2 Parameter	
DiagnosticConnectedIndicator.healingCycleCounterThreshold	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00748]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter
BSW Parameter	**BSW Type**
DemStorageConditionGroupRef	ECUC-REFERENCE-DEF
BSW Description	
References a storage condition group.	
Template Description	
Defines which StorageConditionGroup is applicable for a DiagnosticEvent.	
M2 Parameter	
DiagnosticEventToStorageConditionGroupMapping	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00769]

BSW Module	BSW Context
Dem	Dem/DemConfigSet
BSW Parameter	**BSW Type**
DemMultiEventTriggering	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configures an event that will trigger other events whenever the event is reported.	
Template Description	
This meta-class provides the ability to map a master diagnostic event with a slave diagnostic event such that reporting of the master event with a given value also reports the slave event with the same value	
M2 Parameter	
DiagnosticMasterToSlaveEventMapping	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00944]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemMultiEventTriggering
BSW Parameter	**BSW Type**
DemMultiEventTriggeringMasterEventRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the event that will trigger other events upon reception of this event.	
Template Description	
This represents the master diagnostic event.	
M2 Parameter	
DiagnosticMasterToSlaveEventMapping.masterEvent	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00945]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemMultiEventTriggering
BSW Parameter	**BSW Type**
DemMultiEventTriggeringSlaveEventRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the event that is triggered upon triggering the master event.	
Template Description	
This represents the slave diagnostic event.	
M2 Parameter	
DiagnosticMasterToSlaveEventMapping.slaveEvent	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00946]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemObdDTC
BSW Parameter	**BSW Type**
DemConsiderPtoStatus	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This parameter is TRUE, when the event is affected by the Dem PTO handling.	
Template Description	
This attribute describes the affection of the event by the Dem PTO handling. true: the event is affected by the Dem PTO handling. false: the event is not affected by the Dem PTO handling.	
M2 Parameter	
DiagnosticTroubleCodeObd.considerPtoStatus	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00602]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemObdDTC
BSW Parameter	**BSW Type**
DemDtcValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Unique Diagnostic Trouble Code value for OBD	
Template Description	
Unique Diagnostic Trouble Code value for OBD.	
M2 Parameter	
DiagnosticTroubleCodeObd.obdDTCValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00885]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemObdDTC
BSW Parameter	**BSW Type**
DemDtcValue3Byte	ECUC-INTEGER-PARAM-DEF
BSW Description	
3 Byte OBD DTC value that is used for the UDS Diagnostic Trouble Code (DTC) if UDS and OBD DTC handling is separated. The value is defining a DTC based on SAE J2012.	
Template Description	
3 Byte OBD DTC value based on the definition from SAE J2012. The existence of this attribute is only required if separated UDS and OBD DTC values are used for SAE J1979-2. If this attribute does not exist, then UDS DTC values are used with J1979-2.	
M2 Parameter	
DiagnosticTroubleCodeUds.obdDtcValue3Byte	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00988]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemObdDTC
BSW Parameter	**BSW Type**
DemEventOBDReadinessGroup	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter specifies the Event OBD Readiness group as defined by SAE J1979-DA for PID $01 and PID $41 computation. This parameter is only applicable for emission-related ECUs.	
Template Description	
DiagnosticTroubleCodeObd.eventReadinessGroup: This aggregation allows for the variant definition of the attribute eventObdReadinessGroup. **EventObdReadinessGroup.eventObdReadinessGroup:** This attribute specifies the Event OBD Readiness group for PID $01 and PID $41 computation. This attribute is only applicable for emission-related ECUs. **DiagnosticTroubleCodeUds.eventReadinessGroup:** This attribute specifies the Event OBD Readiness group for PID $01 and PID $41 computation. This attribute is only applicable for emission-related ECUs.	
M2 Parameter	
DiagnosticTroubleCodeObd.eventReadinessGroup, EventObdReadinessGroup.eventObdReadinessGroup, DiagnosticTroubleCodeUds.eventReadinessGroup	
Mapping Rule	**Mapping Type**
Configuration is only possible if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU (see up_Dem_00131) The value of DemEventOBDReadinessGroup can be • derived from DiagnosticTroubleCodeObd.eventReadinessGroup.eventObdReadinessGroup if OBD-II is used, and • derived from DiagnosticTroubleCodeUds.eventReadinessGroup.eventObdReadinessGroup if OBD-on-UDS is used.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00755]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemObdDTC
BSW Parameter	**BSW Type**
DemJ1939DTCValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Unique Diagnostic Trouble Code value for J1939 (consisting of SPN and FMI).	
Template Description	
DiagnosticTroubleCodeJ1939.spn: This represents the releated SPN. **DiagnosticTroubleCodeJ1939.fmi:** This attribute represents the behavior of the Failure Mode Indicator.	
M2 Parameter	
DiagnosticTroubleCodeJ1939.spn, DiagnosticTroubleCodeJ1939.fmi	
Mapping Rule	**Mapping Type**
The value is created out of a combination of the two attribute fmi and spn. The details are explained in the J1939-73 document	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00892]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemPidClass
BSW Parameter	**BSW Type**
DemPidDataElement	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the different data elements contained in the specific PID.	
Template Description	
This represents the data carried by the DiagnosticParameterIdentifier.	
M2 Parameter	
DiagnosticParameterIdentifier.dataElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00896]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemPidClass/DemPidDataElement
BSW Parameter	**BSW Type**
DemPidDataElementClassRef	ECUC-REFERENCE-DEF
BSW Description	
This reference contains the link to a data element class.	
Template Description	
This meta-class represents the ability to describe a concrete piece of data to be taken into account for diagnostic purposes.	
M2 Parameter	
DiagnosticDataElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_Dem_00733]

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemPidClass
BSW Parameter	**BSW Type**
DemPidIdentifier	ECUC-INTEGER-PARAM-DEF
BSW Description	
identifier of the PID	
Template Description	
This is the numerical identifier used to identify the DiagnosticParameterIdentifier in the scope of diagnostic workflow (see SAE J1979-DA).	
M2 Parameter	
DiagnosticParameterIdentifier.id	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00705]

BSW Module	BSW Context
Dem	Dem/DemGeneral
BSW Parameter	**BSW Type**
DemAgingRequiresTestedCycle	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines if the aging cycle counter is processed every aging cycles or if only tested aging cycle are considered. true: only tested aging cycle are considered for aging cycle counter false: aging cycle counter is processed every aging cycle	
Template Description	
Defines whether the aging cycle counter is processed every aging cycles or else only tested aging cycle are considered. If the attribute is set to TRUE: only tested aging cycle are considered for aging cycle counter. If the attribute is set to FALSE: aging cycle counter is processed every aging cycle. On the classic platform, the value of this attribute has to be identical for each DiagnosticMemoryDestination.	
M2 Parameter	
DiagnosticMemoryDestination.agingRequiresTestedCycle	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00877]

BSW Module	BSW Context
Dem	Dem/DemGeneral
BSW Parameter	**BSW Type**
DemClearDTCLimitation	ECUC-ENUMERATION-PARAM-DEF
BSW Description	

▽

△

Defines the supported Dem_<...>ClearDTC API scope.	
Template Description	
Defines the scope of the DEM_ClearDTC Api. On the classic platform, the value of this attribute has to be identical for each DiagnosticMemoryDestination.	
M2 Parameter	
DiagnosticMemoryDestination.clearDtcLimitation	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00790]

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemClearDTCLimitation
BSW Parameter	**BSW Type**
DEM_ALL_SUPPORTED_DTCS	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Dem_<...>ClearDTC accepts all supported DTC values, as well as all DTC values which are configured in DemGroupDTCs and DEM_DTC_GROUP_ALL_DTCS.	
Template Description	
DEM_ClearDtc API accepts all supported DTC values.	
M2 Parameter	
DiagnosticClearDtcLimitationEnum.allSupportedDtcs	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemClearDTCLimitation
BSW Parameter	**BSW Type**
DEM_ONLY_CLEAR_ALL_DTCS	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Dem_<...>ClearDTC accepts ClearAllDTCs only.	
Template Description	
DEM_ClearDtc API accepts ClearAllDTCs only.	
M2 Parameter	
DiagnosticClearDtcLimitationEnum.clearAllDtcs	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
BOOLEAN	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is boolean.

Template Description

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

M2 Parameter

DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type
• baseTypeEncoding = BOOLEAN • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
DOUBLE	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is double (64 bit float).

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type
• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
DOUBLE_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is double (64 bit float) array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
FLOAT	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is float.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type



△

• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType
BSW Parameter	**BSW Type**
FLOAT_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is float array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	

Mapping Rule	**Mapping Type**
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType
BSW Parameter	**BSW Type**
SINT16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint16.	
Template Description	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
• baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
SINT16_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
Type of the data is sint16 array.

Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
• baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
SINT32	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint32.

Template Description

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

M2 Parameter

DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type
• baseTypeEncoding = 2C • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
SINT32_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint32 array.

Template Description

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

M2 Parameter

DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type



△

• baseTypeEncoding = 2C • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType
BSW Parameter	**BSW Type**
SINT64	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint64.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	

Mapping Rule	**Mapping Type**
• baseTypeEncoding = 2C • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType
BSW Parameter	**BSW Type**
SINT64_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint64 array.	
Template Description	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
• baseTypeEncoding = 2C • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
SINT8	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint8.	
Template Description	
DiagnosticDataElement.arraySizeSemantics: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.	
M2 Parameter	
DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize	

Mapping Rule	Mapping Type
• baseTypeEncoding = 2C • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
SINT8_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint8 array.

Template Description

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

M2 Parameter

DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type
• baseTypeEncoding = 2C • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
UINT16	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is uint16.

Template Description

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

M2 Parameter

DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type



△

• baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType

BSW Parameter	**BSW Type**
UINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	

Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType

BSW Parameter	**BSW Type**
UINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint32.	
Template Description	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType

BSW Parameter	**BSW Type**
UINT32_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint32 array.	
Template Description	
DiagnosticDataElement.arraySizeSemantics: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.	
M2 Parameter	
DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize	

Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
UINT64	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is uint64.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
UINT64_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is uint64 array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type

▽

△

• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType
BSW Parameter	**BSW Type**
UINT8	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeSize, BaseTypeDirectDefinition.baseTypeEncoding, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics	

Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElement DataType
BSW Parameter	**BSW Type**
UINT8_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint8 array.	
Template Description	

▽

△

DiagnosticDataElement.arraySizeSemantics: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.
M2 Parameter
DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type
• baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
BOOLEAN	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
Type of the data is boolean.

Template Description
DiagnosticDataElement.arraySizeSemantics: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.
M2 Parameter
DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type
• baseTypeEncoding = BOOLEAN • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
DOUBLE	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is double (64 bit float).

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type
• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
DOUBLE_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is double (64 bit float) array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
• baseTypeEncoding = IEEE754 • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
FLOAT	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is float.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
FLOAT_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is float array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type



△

• baseTypeEncoding = IEEE754 • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType
BSW Parameter	**BSW Type**
SINT16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	

Mapping Rule	**Mapping Type**
• baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType
BSW Parameter	**BSW Type**
SINT16_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint16 array.	
Template Description	

▽

△

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
• baseTypeEncoding = 2C • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
SINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint32.	
Template Description	
DiagnosticDataElement.arraySizeSemantics: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.	
M2 Parameter	
DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize	

Mapping Rule	Mapping Type
• baseTypeEncoding = 2C • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
SINT32_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint32 array.

Template Description

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

M2 Parameter

DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type
• baseTypeEncoding = 2C • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
SINT64	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is sint64.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type

△

• baseTypeEncoding = 2C • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType

BSW Parameter	**BSW Type**
SINT64_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint64 array.	
Template Description	

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = 2C • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType

BSW Parameter	**BSW Type**
SINT8	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is sint8.	
Template Description	

▽

△

DiagnosticDataElement.arraySizeSemantics: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.
M2 Parameter
DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	**Mapping Type**
• baseTypeEncoding = 2C • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType

BSW Parameter	**BSW Type**
SINT8_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
Type of the data is sint8 array.
Template Description
DiagnosticDataElement.arraySizeSemantics: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.
M2 Parameter
DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	**Mapping Type**
• baseTypeEncoding = 2C • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
UINT16	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is uint16.

Template Description

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

M2 Parameter

DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type
• baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
UINT16_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is uint16 array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type

▽

△

• baseTypeEncoding = NONE • baseTypeSize = 16 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType
BSW Parameter	**BSW Type**
UINT32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 32 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType
BSW Parameter	**BSW Type**
UINT32_N	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Type of the data is uint32 array.	
Template Description	

▽

△

DiagnosticDataElement.arraySizeSemantics: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.
M2 Parameter
DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 32 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType

BSW Parameter	**BSW Type**
UINT64	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
Type of the data is uint64.

Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics:** This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
UINT64_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is uint64 array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize, DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements

Mapping Rule	Mapping Type
• baseTypeEncoding = NONE • baseTypeSize = 64 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType

BSW Parameter	BSW Type
UINT8	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

Type of the data is uint8.

Template Description

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

BaseTypeDirectDefinition.baseTypeSize, BaseTypeDirectDefinition.baseTypeEncoding, DiagnosticDataElement.maxNumberOfElements, DiagnosticDataElement.arraySizeSemantics

Mapping Rule	Mapping Type

△

• baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements shall not exist • arraySizeSemantics shall not exist	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElement DataType

BSW Parameter	**BSW Type**
UINT8_N	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
Type of the data is uint8 array.

Template Description
DiagnosticDataElement.arraySizeSemantics: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements:** The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **BaseTypeDirectDefinition.baseTypeEncoding:** This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize:** Describes the length of the data type specified in the container in bits.

M2 Parameter
DiagnosticDataElement.arraySizeSemantics, DiagnosticDataElement.maxNumberOfElements, BaseTypeDirectDefinition.baseTypeEncoding, BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	**Mapping Type**
• baseTypeEncoding = NONE • baseTypeSize = 8 • maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) • arraySizeSemantics either does not exist or exists and is set to [ARMetaClassRole{fixed Size}{ArraySizeSemanticsEnum} (cf. TPS_DEXT_01001)	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemInternalDataElementClass

BSW Parameter	**BSW Type**
DemInternalDataElement	ECUC-ENUMERATION-PARAM-DEF

BSW Description
This parameter defines the Dem-internal data value, which is mapped to the data element.

Template Description
This meta-class represents an element of an extended data record. Such an element represents a primitive data object or a fixed-size array of primitive data objects without further internal structure (no structure allowed).

M2 Parameter
DiagnosticExtendedDataRecordElement

Mapping Rule	**Mapping Type**

▽

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00616]

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDataElementClass/DemInternalDataElementClass/DemInternalData Element
BSW Parameter	**BSW Type**
DEM_EVENT_ASSOCIATED_IDENTIFICATION	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Represents the static value associated to an event by DemEventAssociatedIdentification.	
Template Description	
This attribute represents the identification number that is associated with the enclosing DiagnosticEvent and allows to identify it when placed into a snapshot record or extended data record storage. This value can be reported as internal data element in snapshot records or extended data records.	
M2 Parameter	
DiagnosticEvent.associatedEventIdentification	
Mapping Rule	**Mapping Type**
If attribute DiagnosticEvent.associatedEventIdentification exists then the corresponding Dem InternalDataElement shall be set to DEM_EVENT_ASSOCIATED_IDENTIFICATION.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral
BSW Parameter	**BSW Type**
DemDidClass	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration (parameters) for a data Id class. It is assembled out of one or several data elements.	
Template Description	
This is the numerical identifier used to identify the DiagnosticAbstractDataIdentifier in the scope of diagnostic workflow	
M2 Parameter	
DiagnosticAbstractDataIdentifier.id	
Mapping Rule	**Mapping Type**
If the subclass of DiagnosticAbstractDataIdentifier is referenced as DiagnosticTroubleCode Props.freezeFrameContent.dataIdentifier	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00706]

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemDidClass
BSW Parameter	**BSW Type**
DemDidIdentifier	ECUC-INTEGER-PARAM-DEF
BSW Description	
Identifier of the Data ID.	

▽

△

Template Description	
This is the numerical identifier used to identify the DiagnosticAbstractDataIdentifier in the scope of diagnostic workflow	
M2 Parameter	
DiagnosticAbstractDataIdentifier.id	
Mapping Rule	**Mapping Type**
If the DiagnosticDataIdentifier is referenced as DiagnosticTroubleCodeProps.freezeFrame Content.dataIdentifier	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00650]

BSW Module	BSW Context
Dem	Dem/DemGeneral
BSW Parameter	**BSW Type**
DemEnableCondition	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration (parameters) for enable conditions.	
Template Description	
Specification of an enable condition.	
M2 Parameter	
DiagnosticEnableCondition	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00653]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEnableCondition
BSW Parameter	**BSW Type**
DemEnableConditionStatus	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines the initial status for enable or disable of acceptance of event reports of a diagnostic event. The value is the initialization after power up (before this condition is reported the first time). true: acceptance of a diagnostic event enabled false: acceptance of a diagnostic event disabled	
Template Description	
Defines the initial status for enable or disable of acceptance/storage of event reports of a diagnostic event. The value is the initialization after power up (before this condition is reported the first time). true: acceptance/storage of a diagnostic event enabled false: acceptance/storage of a diagnostic event disabled	
M2 Parameter	
DiagnosticCondition.initValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00656]

BSW Module	BSW Context	
Dem	Dem/DemGeneral	
BSW Parameter		**BSW Type**
DemEnableConditionGroup		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container contains the configuration (parameters) for enable condition groups.		
Template Description		
Enable condition group which includes one or several enable conditions.		
M2 Parameter		
DiagnosticEnableConditionGroup		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dem_00745]

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemEnableConditionGroup	
BSW Parameter		**BSW Type**
DemEnableConditionRef		ECUC-REFERENCE-DEF
BSW Description		
References an enable condition.		
Template Description		
Reference to enableConditions that are part of the EnableConditionGroup.		
M2 Parameter		
DiagnosticEnableConditionGroup.enableCondition		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dem_00655]

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemEventMemorySet	
BSW Parameter		**BSW Type**
DemIndicator		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container contains the configuration (parameters) for Indicators.		
Template Description		
Definition of an indicator.		
M2 Parameter		
DiagnosticIndicator		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dem_00680]

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemEventMemorySet	
BSW Parameter	**BSW Type**	
DemMILIndicatorRef	ECUC-REFERENCE-DEF	
BSW Description		
This parameter defines the indicator representing the MIL. This parameter is mandatory for ECUs supporting OBD (refer to DemOBDSupport).		
Template Description		
Defines the type of the indicator.		
M2 Parameter		
DiagnosticIndicator.type		
Mapping Rule		**Mapping Type**
DiagnosticIndicator.type == DiagnosticIndicatorTypeEnum.malfunction		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dem_00723]

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemEventMemorySet	
BSW Parameter	**BSW Type**	
DemPrimaryMemory	ECUC-PARAM-CONF-CONTAINER-DEF	
BSW Description		
This container contains the primary event memory specific parameters of the Dem module.		
Template Description		
This represents a primary memory for a diagnostic event.		
M2 Parameter		
DiagnosticMemoryDestinationPrimary		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dem_00901]

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory	
BSW Parameter	**BSW Type**	
DemDtcStatusAvailabilityMask	ECUC-INTEGER-PARAM-DEF	
BSW Description		
Mask for the supported DTC status bits by the Dem. This mask is used in the positive response of UDS service 0x19.		
Template Description		
Mask for the supported DTC status bits by the Dem.		
M2 Parameter		
DiagnosticMemoryDestination.dtcStatusAvailabilityMask		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dem_00978]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory
BSW Parameter	**BSW Type**
DemEventDisplacementStrategy	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This configuration switch defines, whether support for event displacement is enabled or not, and which displacement strategy is followed.	
Template Description	
This attribute defines, whether support for event displacement is enabled or not, and which displacement strategy is followed.	
M2 Parameter	
DiagnosticMemoryDestination.eventDisplacementStrategy	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00980]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory/DemEventDisplacementStrategy
BSW Parameter	**BSW Type**
DEM_DISPLACEMENT_FULL	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Event memory entry displacement is enabled, by consideration of priority active/passive status, and occurrence.	
Template Description	
Event memory entry displacement is enabled, by consideration of priority active/passive status, and occurrence.	
M2 Parameter	
DiagnosticEventDisplacementStrategyEnum.full	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory/DemEventDisplacementStrategy
BSW Parameter	**BSW Type**
DEM_DISPLACEMENT_NONE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Event memory entry displacement is disabled.	
Template Description	
Event memory entry displacement is disabled.	
M2 Parameter	
DiagnosticEventDisplacementStrategyEnum.none	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory/DemEventDisplacementStrategy
BSW Parameter	**BSW Type**
DEM_DISPLACEMENT_PRIO_OCC	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Event memory entry displacement is enabled, by consideration of priority and occurrence (but without active/passive status).	
Template Description	
Event memory entry displacement is enabled, by consideration of priority and occurrence (but without active/passive status).	
M2 Parameter	
DiagnosticEventDisplacementStrategyEnum.prioOcc	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory
BSW Parameter	**BSW Type**
DemEventMemoryEntryStorageTrigger	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Configures the primary trigger to allocate an event memory entry.	
Template Description	
Describes the trigger to allocate an event memory entry.	
M2 Parameter	
DiagnosticMemoryDestination.memoryEntryStorageTrigger	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00982]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory/DemEventMemoryEntryStorage Trigger
BSW Parameter	**BSW Type**
DEM_TRIGGER_ON_CONFIRMED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Event Memory entries are triggered if the UDS status bit 3 (confirmedDTC) changes from 0 to 1.	
Template Description	
Status information of UDS DTC status bit 3	
M2 Parameter	
DiagnosticMemoryEntryStorageTriggerEnum.confirmed	
Mapping Rule	**Mapping Type**

△

DemPrimaryMemory.DemEventMemoryEntryStorageTrigger.DEM_TRIGGER_ON_CONFIRMEDI maps to DiagnosticMemoryDestinationPrimary.memoryEntryStorageTrigger if the attribute is set to DiagnosticMemoryEntryStorageTriggerEnum.confirmed DemUserDefinedMemory.DemEventMemoryEntryStorageTrigger.DEM_TRIGGER_ON_CONFIRMEDI maps to DiagnosticMemoryDestinationUserDefined.memoryEntryStorageTrigger if the attribute is set toDiagnosticMemoryEntryStorageTriggerEnum.confirmed	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory/DemEventMemoryEntryStorageTrigger
BSW Parameter	**BSW Type**
DEM_TRIGGER_ON_FDC_THRESHOLD	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Event Memory entries are triggered when the FDC threshold is reached.	
Template Description	
Threshold to allocate an event memory entry and to capture the Freeze Frame.	
M2 Parameter	
DiagnosticMemoryEntryStorageTriggerEnum.fdcThreshold	
Mapping Rule	**Mapping Type**
1:1 mapping rules: DemPrimaryMemory.DemEventMemoryEntryStorageTrigger.DEM_TRIGGER_ON_FDC_THRESHOLD maps to DiagnosticMemoryDestinationPrimary.memoryEntryStorageTrigger if the attribute is set to DiagnosticMemoryEntryStorageTriggerEnum.fdcThreshold DemUserDefinedMemory.DemEventMemoryEntryStorageTrigger.DEM_TRIGGER_ON_FDC_THRESHOLD maps to DiagnosticMemoryDestinationUserDefined.memoryEntryStorageTrigger if the attribute is set to DiagnosticMemoryEntryStorageTriggerEnum.fdcThreshold	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory/DemEventMemoryEntryStorageTrigger
BSW Parameter	**BSW Type**
DEM_TRIGGER_ON_TEST_FAILED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Event Memory entries are triggered if the UDS status bit 0 (testFailed) changes from 0 to 1.	
Template Description	
Status information of UDS DTC status bit 0.	
M2 Parameter	
DiagnosticMemoryEntryStorageTriggerEnum.testFailed	
Mapping Rule	**Mapping Type**

▽

△

DemPrimaryMemory.DemEventMemoryEntryStorageTrigger.DEM_TRIGGER_ON_TEST_FAILED maps to DiagnosticMemoryDestinationPrimary.memoryEntryStorageTrigger if the attribute is set to DiagnosticMemoryEntryStorageTriggerEnum.testFailed DemUserDefinedMemory.DemEventMemoryEntryStorageTrigger.DEM_TRIGGER_ON_TEST_FAILED maps to DiagnosticMemoryDestinationUserDefined.memoryEntryStorageTrigger if the attribute is set to DiagnosticMemoryEntryStorageTriggerEnum.testFailed	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory
BSW Parameter	**BSW Type**
DemMaxNumberEventEntryPrimary	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of events which can be stored in the primary memory	
Template Description	
This attribute fixes the maximum number of event entries in the fault memory.	
M2 Parameter	
DiagnosticMemoryDestination.maxNumberOfEventEntries	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00690]

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory
BSW Parameter	**BSW Type**
DemOccurrenceCounterProcessing	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This configuration switch defines the consideration of the fault confirmation process for the occurrence counter. For OBD and mixed systems (OBD/non OBD, refer to DemOBDSupport) configuration switch shall always set to DEM_PROCESS_OCCCTR_TF.	
Template Description	
This attribute defines the consideration of the fault confirmation process for the occurrence counter.	
M2 Parameter	
DiagnosticCommonProps.occurrenceCounterProcessing	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00984]

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory/DemOccurrenceCounter Processing
BSW Parameter	**BSW Type**
DEM_PROCESS_OCCCTR_CDTC	ECUC-ENUMERATION-LITERAL-DEF

▽

△

BSW Description	
the occurrence counter is triggered by the TestFailed bit if the fault confirmation was successful (ConfirmedDTC bit is set)	
Template Description	
The occurrence counter is incremented when TestFailed bit transitions from 0 to 1 if the fault confirmation was successful (ConfirmedDTC bit is already set).	
M2 Parameter	
DiagnosticOccurrenceCounterProcessingEnum.confirmedDtcBit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory/DemOccurrenceCounterProcessing
BSW Parameter	**BSW Type**
DEM_PROCESS_OCCCTR_TF	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
the occurrence counter is only triggered by the TestFailed bit (and the fault confirmation is not considered) This parameter is mandatory in case of J1939.	
Template Description	
The occurrence counter is incremented when TestFailed bit transitions from 0 to 1 (and the fault confirmation is not considered).	
M2 Parameter	
DiagnosticOccurrenceCounterProcessingEnum.testFailedBit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory
BSW Parameter	**BSW Type**
DemTypeOfFreezeFrameRecordNumeration	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines the type of assigning freeze frame record numbers for event-specific freeze frame records.	
Template Description	
This attribute defines the type of assigning freeze frame record numbers for event-specific freeze frame records.	
M2 Parameter	
DiagnosticMemoryDestination.typeOfFreezeFrameRecordNumeration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00986]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory/DemTypeOfFreezeFrameRecord Numeration
BSW Parameter	**BSW Type**
DEM_FF_RECNUM_CALCULATED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
freeze frame records will be numbered consecutive starting by 1 in their chronological order	
Template Description	
Freeze frame records will be numbered consecutive starting by 1 in their chronological order.	
M2 Parameter	
DiagnosticTypeOfFreezeFrameRecordNumerationEnum.calculated	
Mapping Rule	**Mapping Type**
If DiagnosticMemoryDestination.typeOfFreezeFrameRecordNumeration is set to DiagnosticTypeOf FreezeFrameRecordNumerationEnum.calculated	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory/DemTypeOfFreezeFrameRecord Numeration
BSW Parameter	**BSW Type**
DEM_FF_RECNUM_CONFIGURED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
freeze frame records will be numbered based on the given configuration in their chronological order	
Template Description	
Freeze frame records will be numbered based on the given configuration in their chronological order.	
M2 Parameter	
DiagnosticTypeOfFreezeFrameRecordNumerationEnum.configured	
Mapping Rule	**Mapping Type**
If DiagnosticMemoryDestination.typeOfFreezeFrameRecordNumeration is set to DiagnosticTypeOf FreezeFrameRecordNumerationEnum.configured.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet
BSW Parameter	**BSW Type**
DemTypeOfDTCSupported	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines the format returned by Dem_GetTranslationType and does not relate to/influence the supported Dem functionality.	
Template Description	
This attribute defines the format returned by Dem_DcmGetTranslationType and does not relate to/influence the supported Dem functionality.	
M2 Parameter	
DiagnosticMemoryDestinationPrimary.typeOfDtcSupported	
Mapping Rule	**Mapping Type**



△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00720]

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet/DemTypeOfDTCSupported
BSW Parameter	**BSW Type**
DEM_DTC_TRANSLATION_ISO11992_4	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
ISO11992-4 DTC format	
Template Description	
ISO11992-4 DTC format	
M2 Parameter	
DiagnosticTypeOfDtcSupportedEnum.iso11992_4	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet/DemTypeOfDTCSupported
BSW Parameter	**BSW Type**
DEM_DTC_TRANSLATION_ISO14229_1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
ISO14229-1 DTC format (3 byte format)	
Template Description	
ISO14229-1 DTC format (3 byte format)	
M2 Parameter	
DiagnosticTypeOfDtcSupportedEnum.iso14229_1	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet/DemTypeOfDTCSupported
BSW Parameter	**BSW Type**
DEM_DTC_TRANSLATION_ISO15031_6	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
ISO15031-6 DTC format = SAE_J2012-DA_DTCFormat_00 (2 byte format)	
Template Description	
ISO 15031-6 DTC format (2 byte format) = SAE J2012-DA_DTCFormat_00 (2 Byte format)	
M2 Parameter	
DiagnosticTypeOfDtcSupportedEnum.iso15031_6	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemTypeOfDTCSupported
BSW Parameter	**BSW Type**
DEM_DTC_TRANSLATION_SAEJ1939_73	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
SAEJ1939-73 DTC format	
Template Description	
SAEJ1939-73 DTC format	
M2 Parameter	
DiagnosticTypeOfDtcSupportedEnum.saeJ1939_73	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemTypeOfDTCSupported
BSW Parameter	**BSW Type**
DEM_DTC_TRANSLATION_SAE_J2012_DA_ DTCFORMAT_04	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
SAE_J2012-DA_DTCFormat_00 (3 byte format)	
Template Description	
SAE_J2012-DA_DTCFormat_00 (3 byte format)	
M2 Parameter	
DiagnosticTypeOfDtcSupportedEnum.saeJ2012_da	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet
BSW Parameter	**BSW Type**
DemUserDefinedMemory	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the user defined event memory specific parameters of the Dem module.	
Template Description	
This represents a user-defined memory for a diagnostic event.	

▽

△

M2 Parameter	
DiagnosticMemoryDestinationUserDefined	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00910]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory
BSW Parameter	**BSW Type**
DemDtcStatusAvailabilityMask	ECUC-INTEGER-PARAM-DEF
BSW Description	
Mask for the supported DTC status bits by the Dem. This mask is used in the positive response of UDS service 0x19.	
Template Description	
Mask for the supported DTC status bits by the Dem.	
M2 Parameter	
DiagnosticMemoryDestination.dtcStatusAvailabilityMask	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00978]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory
BSW Parameter	**BSW Type**
DemEventDisplacementStrategy	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This configuration switch defines, whether support for event displacement is enabled or not, and which displacement strategy is followed.	
Template Description	
This attribute defines, whether support for event displacement is enabled or not, and which displacement strategy is followed.	
M2 Parameter	
DiagnosticMemoryDestination.eventDisplacementStrategy	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00980]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory/DemEventDisplacement Strategy
BSW Parameter	**BSW Type**
DEM_DISPLACEMENT_FULL	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Event memory entry displacement is enabled, by consideration of priority active/passive status, and occurrence.	
Template Description	
Event memory entry displacement is enabled, by consideration of priority active/passive status, and occurrence.	
M2 Parameter	
DiagnosticEventDisplacementStrategyEnum.full	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory/DemEventDisplacement Strategy
BSW Parameter	**BSW Type**
DEM_DISPLACEMENT_NONE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Event memory entry displacement is disabled.	
Template Description	
Event memory entry displacement is disabled.	
M2 Parameter	
DiagnosticEventDisplacementStrategyEnum.none	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory/DemEventDisplacement Strategy
BSW Parameter	**BSW Type**
DEM_DISPLACEMENT_PRIO_OCC	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Event memory entry displacement is enabled, by consideration of priority and occurrence (but without active/passive status).	
Template Description	
Event memory entry displacement is enabled, by consideration of priority and occurrence (but without active/passive status).	
M2 Parameter	
DiagnosticEventDisplacementStrategyEnum.prioOcc	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory
BSW Parameter	**BSW Type**

▽

△

DemEventMemoryEntryStorageTrigger	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Configures the primary trigger to allocate an event memory entry.	
Template Description	
Describes the trigger to allocate an event memory entry.	
M2 Parameter	
DiagnosticMemoryDestination.memoryEntryStorageTrigger	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00982]

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory/DemEventMemoryEntry StorageTrigger
BSW Parameter	**BSW Type**
DEM_TRIGGER_ON_CONFIRMED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Event Memory entries are triggered if the UDS status bit 3 (confirmedDTC) changes from 0 to 1.	
Template Description	
Status information of UDS DTC status bit 3	
M2 Parameter	
DiagnosticMemoryEntryStorageTriggerEnum.confirmed	
Mapping Rule	**Mapping Type**
DemPrimaryMemory.DemEventMemoryEntryStorageTrigger.DEM_TRIGGER_ON_CONFIRMEDI maps to DiagnosticMemoryDestinationPrimary.memoryEntryStorageTrigger if the attribute is set to DiagnosticMemoryEntryStorageTriggerEnum.confirmed DemUserDefinedMemory.DemEventMemoryEntryStorageTrigger.DEM_TRIGGER_ON_ CONFIRMEDI maps to DiagnosticMemoryDestinationUserDefined.memoryEntryStorageTrigger if the attribute is set toDiagnosticMemoryEntryStorageTriggerEnum.confirmed	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory/DemEventMemoryEntry StorageTrigger
BSW Parameter	**BSW Type**
DEM_TRIGGER_ON_FDC_THRESHOLD	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Event Memory entries are triggered when the FDC threshold is reached.	
Template Description	
Threshold to allocate an event memory entry and to capture the Freeze Frame.	
M2 Parameter	
DiagnosticMemoryEntryStorageTriggerEnum.fdcThreshold	
Mapping Rule	**Mapping Type**

▽

△

1:1 mapping rules: DemPrimaryMemory.DemEventMemoryEntryStorageTrigger.DEM_TRIGGER_ON_FDC_THRESHOLD maps to DiagnosticMemoryDestinationPrimary.memoryEntryStorageTrigger if the attribute is set to DiagnosticMemoryEntryStorageTriggerEnum.fdcThreshold DemUserDefinedMemory.DemEventMemoryEntryStorageTrigger.DEM_TRIGGER_ON_FDC_THRESHOLD maps to DiagnosticMemoryDestinationUserDefined.memoryEntryStorageTrigger if the attribute is set to DiagnosticMemoryEntryStorageTriggerEnum.fdcThreshold	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory/DemEventMemoryEntryStorageTrigger
BSW Parameter	**BSW Type**
DEM_TRIGGER_ON_TEST_FAILED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Event Memory entries are triggered if the UDS status bit 0 (testFailed) changes from 0 to 1.	
Template Description	
Status information of UDS DTC status bit 0.	
M2 Parameter	
DiagnosticMemoryEntryStorageTriggerEnum.testFailed	

Mapping Rule	**Mapping Type**
DemPrimaryMemory.DemEventMemoryEntryStorageTrigger.DEM_TRIGGER_ON_TEST_FAILED maps to DiagnosticMemoryDestinationPrimary.memoryEntryStorageTrigger if the attribute is set to DiagnosticMemoryEntryStorageTriggerEnum.testFailed DemUserDefinedMemory.DemEventMemoryEntryStorageTrigger.DEM_TRIGGER_ON_TEST_FAILED maps to DiagnosticMemoryDestinationUserDefined.memoryEntryStorageTrigger if the attribute is set to DiagnosticMemoryEntryStorageTriggerEnum.testFailed	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory
BSW Parameter	**BSW Type**
DemMaxNumberEventEntryUserDefined	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of events which can be stored in the user defined memory.	
Template Description	
This attribute fixes the maximum number of event entries in the fault memory.	
M2 Parameter	
DiagnosticMemoryDestination.maxNumberOfEventEntries	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00691]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory
BSW Parameter	**BSW Type**
DemOccurrenceCounterProcessing	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This configuration switch defines the consideration of the fault confirmation process for the occurrence counter. For OBD and mixed systems (OBD/non OBD, refer to DemOBDSupport) configuration switch shall always set to DEM_PROCESS_OCCCTR_TF.	
Template Description	
This attribute defines the consideration of the fault confirmation process for the occurrence counter.	
M2 Parameter	
DiagnosticCommonProps.occurrenceCounterProcessing	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00984]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory/DemOccurrenceCounterProcessing
BSW Parameter	**BSW Type**
DEM_PROCESS_OCCCTR_CDTC	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
the occurrence counter is triggered by the TestFailed bit if the fault confirmation was successful (ConfirmedDTC bit is set)	
Template Description	
The occurrence counter is incremented when TestFailed bit transitions from 0 to 1 if the fault confirmation was successful (ConfirmedDTC bit is already set).	
M2 Parameter	
DiagnosticOccurrenceCounterProcessingEnum.confirmedDtcBit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory/DemOccurrenceCounterProcessing
BSW Parameter	**BSW Type**
DEM_PROCESS_OCCCTR_TF	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
the occurrence counter is only triggered by the TestFailed bit (and the fault confirmation is not considered) This parameter is mandatory in case of J1939.	
Template Description	
The occurrence counter is incremented when TestFailed bit transitions from 0 to 1 (and the fault confirmation is not considered).	
M2 Parameter	
DiagnosticOccurrenceCounterProcessingEnum.testFailedBit	

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory
BSW Parameter	**BSW Type**
DemTypeOfFreezeFrameRecordNumeration	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines the type of assigning freeze frame record numbers for event-specific freeze frame records.	
Template Description	
This attribute defines the type of assigning freeze frame record numbers for event-specific freeze frame records.	
M2 Parameter	
DiagnosticMemoryDestination.typeOfFreezeFrameRecordNumeration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00986]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory/DemTypeOfFreezeFrame RecordNumeration
BSW Parameter	**BSW Type**
DEM_FF_RECNUM_CALCULATED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
freeze frame records will be numbered consecutive starting by 1 in their chronological order	
Template Description	
Freeze frame records will be numbered consecutive starting by 1 in their chronological order.	
M2 Parameter	
DiagnosticTypeOfFreezeFrameRecordNumerationEnum.calculated	
Mapping Rule	**Mapping Type**
If DiagnosticMemoryDestination.typeOfFreezeFrameRecordNumeration is set to DiagnosticTypeOf FreezeFrameRecordNumerationEnum.calculated	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory/DemTypeOfFreezeFrame RecordNumeration
BSW Parameter	**BSW Type**
DEM_FF_RECNUM_CONFIGURED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
freeze frame records will be numbered based on the given configuration in their chronological order	
Template Description	

▽

△

Freeze frame records will be numbered based on the given configuration in their chronological order.	
M2 Parameter	
DiagnosticTypeOfFreezeFrameRecordNumerationEnum.configured	
Mapping Rule	**Mapping Type**
If DiagnosticMemoryDestination.typeOfFreezeFrameRecordNumeration is set to DiagnosticTypeOf FreezeFrameRecordNumerationEnum.configured.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory
BSW Parameter	**BSW Type**
DemUserDefinedMemoryIdentifier	ECUC-INTEGER-PARAM-DEF
BSW Description	
Identifier used by external tester to identify the User defined event memory.	
Template Description	
This represents the identifier of the user-defined memory.	
M2 Parameter	
DiagnosticMemoryDestinationUserDefined.memoryId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00903]

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemExtendedDataRecordClass
BSW Parameter	**BSW Type**
DemExtendedDataRecordNumber	ECUC-INTEGER-PARAM-DEF
BSW Description	
This configuration parameter specifies a number for an extended data record. The value is unique within a fault memory. One or more extended data records can be assigned to one diagnostic event/DTC. For primary fault memory, the value 0x00 and the range from 0xF0 to 0xFF are either reserved or predefined. For user defined fault memories, only the value 0x00 is reserved and 0xFF is predefined.	
Template Description	
This attribute specifies an unique identifier for an extended data record.	
M2 Parameter	
DiagnosticExtendedDataRecord.recordNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00666]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemExtendedDataRecordClass
BSW Parameter	**BSW Type**
DemExtendedDataRecordTrigger	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the trigger to store the ExtendedDataRecord.	
Template Description	
This attribute specifies the primary trigger to allocate an event memory entry.	
M2 Parameter	
DiagnosticExtendedDataRecord.trigger	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00804]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemExtendedDataRecordClass/DemExtendedDataRecordTrigger
BSW Parameter	**BSW Type**
DEM_TRIGGER_ON_CONFIRMED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
ExtendedDataRecord will be stored when the UDS status confirmed bit changes from 0 to 1.	
Template Description	
capture on "Confirmed"	
M2 Parameter	
DiagnosticRecordTriggerEnum.confirmed	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemExtendedDataRecordClass/DemExtendedDataRecordTrigger
BSW Parameter	**BSW Type**
DEM_TRIGGER_ON_FDC_THRESHOLD	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
ExtendedDataRecord will be stored when the FDC reaches its threshold.	
Template Description	
capture on "FDC Threshold"	
M2 Parameter	
DiagnosticRecordTriggerEnum.fdcThreshold	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemExtendedDataRecordClass/DemExtendedDataRecordTrigger
BSW Parameter	**BSW Type**
DEM_TRIGGER_ON_PASSED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
ExtendedDataRecord will be stored when the event is reported as passed (testFailed bit transition 1 --> 0).	
Template Description	
Capture on testFailed bit transition 1 --> 0.	
M2 Parameter	
DiagnosticRecordTriggerEnum.testPassed	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemExtendedDataRecordClass/DemExtendedDataRecordTrigger
BSW Parameter	**BSW Type**
DEM_TRIGGER_ON_PENDING	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
ExtendedDataRecord will be stored when the UDS status pending bit changes from 0 to 1.	
Template Description	
capture on "Pending"	
M2 Parameter	
DiagnosticRecordTriggerEnum.pending	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemExtendedDataRecordClass/DemExtendedDataRecordTrigger
BSW Parameter	**BSW Type**
DEM_TRIGGER_ON_TEST_FAILED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
ExtendedDataRecord will be stored when the UDS status test failed bit changes from 0 to 1.	
Template Description	
capture on "Test Failed"	
M2 Parameter	
DiagnosticRecordTriggerEnum.testFailed	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemExtendedDataRecordClass
BSW Parameter	**BSW Type**
DemExtendedDataRecordUpdate	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This extended data record is captured if the configured trigger condition in "DemExtendedDataRecordTrigger" is fulfilled.	
Template Description	
This attribute defines when an extended data record is captured. true: This extended data record is captured every time. false: This extended data record is only captured for new event memory entries.	
M2 Parameter	
DiagnosticExtendedDataRecord.update	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00621]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemFreezeFrameRecordClass
BSW Parameter	**BSW Type**
DemFreezeFrameRecordNumber	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the snapshot record number for a freeze frame. This record number is unique per freeze frame record number class. The number needs to be unique within a fault memory.	
Template Description	
This attribute defines a record number for a freeze frame record.	
M2 Parameter	
DiagnosticFreezeFrame.recordNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00777]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemFreezeFrameRecordClass
BSW Parameter	**BSW Type**
DemFreezeFrameRecordTrigger	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the trigger to store the FreezeFrameRecord.	
Template Description	
This attribute defines the primary trigger to allocate an event memory entry.	
M2 Parameter	
DiagnosticFreezeFrame.trigger	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
draft	[ECUC_Dem_00803]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemFreezeFrameRecordClass/DemFreezeFrameRecordTrigger
BSW Parameter	**BSW Type**
DEM_TRIGGER_ON_TEST_FAILED_THIS_OPERATION_CYCLE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Event Memory entries are triggered if the UDS status bit 1 (testFailedThisOperationCycle) changes from 0 to 1.	
Template Description	
Test Failed This Operation Cycle.	
M2 Parameter	
DiagnosticRecordTriggerEnum.testFailedThisOperationCycle	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemFreezeFrameRecordClass
BSW Parameter	**BSW Type**
DemFreezeFrameRecordUpdate	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines the case, when the freeze frame record is stored/updated.	
Template Description	
This attribute defines the approach when the freeze frame record is stored/updated. true: FreezeFrame record is captured every time. false: FreezeFrame record is only captured for new event memory entries.	
M2 Parameter	
DiagnosticFreezeFrame.update	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00802]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemGeneralOBD
BSW Parameter	**BSW Type**
DemSupportedObdUdsDtcSeparation	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If this parameter is set to true, the OBD UDS DTC separation is enabled. If this parameter is set to false, the OBD UDS DTC separation is disabled.	
Template Description	

▽

△

3 Byte OBD DTC value based on the definition from SAE J2012. The existence of this attribute is only required if separated UDS and OBD DTC values are used for SAE J1979-2. If this attribute does not exist, then UDS DTC values are used with J1979-2.	
M2 Parameter	
DiagnosticTroubleCodeUds.obdDtcValue3Byte	
Mapping Rule	**Mapping Type**
DemSupportedObdUdsDtcSeparation shall exist and set to true if, in the context of an enclosing DiagnosticContributionSet, at least one DiagnosticTroubleCodeObd exists where attribute obd DTCValue3Byte exists, independently of its value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00989]

BSW Module	**BSW Context**
Dem	Dem/DemGeneral
BSW Parameter	**BSW Type**
DemOBDSupport	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This configuration switch defines OBD support and kind of OBD ECU.	
Template Description	
This attribute is used to specify the role (if applicable) in which the DiagnosticEcuInstance supports OBD.	
M2 Parameter	
DiagnosticEcuInstanceProps.obdSupport	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00698]

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemOBDSupport
BSW Parameter	**BSW Type**
DEM_OBD_DEP_SEC_ECU	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Kind of OBD ECU: OBD Dependend / Secondary ECU	
Template Description	
This represents the role "secondary ECU".	
M2 Parameter	
DiagnosticObdSupportEnum.secondaryEcu	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemOBDSupport
BSW Parameter	**BSW Type**

▽

△

DEM_OBD_MASTER_ECU	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Kind of OBD ECU: Master ECU	
Template Description	
This represent the role "master ECU".	
M2 Parameter	
DiagnosticObdSupportEnum.masterEcu	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemOBDSupport
BSW Parameter	**BSW Type**
DEM_OBD_NO_OBD_SUPPORT	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
OBD is not supported within this ECU	
Template Description	
This represents the ability to explicitly specify that no participation in OBD is foreseen.	
M2 Parameter	
DiagnosticObdSupportEnum.noObdSupport	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemOBDSupport
BSW Parameter	**BSW Type**
DEM_OBD_PRIMARY_ECU	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Kind of OBD ECU: Pimary ECU	
Template Description	
This represents the role "primary ECU".	
M2 Parameter	
DiagnosticObdSupportEnum.primaryEcu	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context	
Dem	Dem/DemGeneral	
BSW Parameter		**BSW Type**
DemOperationCycle		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container holds all parameters that are relevant to configure an operation cycle.		
Template Description		
Definition of an operation cycle that is the base of the event qualifying and for Dem scheduling.		
M2 Parameter		
DiagnosticOperationCycle		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dem_00701]

BSW Module	BSW Context	
Dem	Dem/DemGeneral	
BSW Parameter		**BSW Type**
DemRatio		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container contains the OBD-specific in-use-monitor performance ratio configuration. It is related to a specific event, a FID, and an IUMPR group.		
Template Description		
ObdRatioServiceNeeds: Specifies the abstract needs of a component or module on the configuration of OBD Services in relation to a particular "ratio monitoring" which is supported by this component or module. **DiagnosticIumprGroup:** This meta-class represents the ability to model a IUMPR groups.		
M2 Parameter		
ObdRatioServiceNeeds, DiagnosticIumprGroup		
Mapping Rule		**Mapping Type**
In case the owner of the ObdRatioServiceNeeds is a BSW module then the DemRatio.shortName = {capitalizedMip}_{ServiceDependency.symbolicNameProps.symbol}. For the DiagnosticIumprGroup the mapping rule is 1:1		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dem_00734]

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemRatio	
BSW Parameter		**BSW Type**
DemDiagnosticEventRef		ECUC-REFERENCE-DEF
BSW Description		
This reference contains the link to a diagnostic event.		
Template Description		

▽

△

DiagnosticIumprGroup.iumpr: This reference collects DiagnosticIumpr to a DiagnosticIumprGroup. **DiagnosticIumpr.event:** This reference represents the DiagnosticEvent that corresponds to the IUMPR computation.	
M2 Parameter	
DiagnosticIumprGroup.iumpr, DiagnosticIumpr.event	
Mapping Rule	**Mapping Type**
foreach DiagnosticIumprGroup, follow the iumpr reference and then pick the target of the event reference	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00735]

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio
BSW Parameter	**BSW Type**
DemFunctionIdRef	ECUC-REFERENCE-DEF
BSW Description	
This reference contains the link to a function identifier within the FiM which is used as a primary FID.	
Template Description	
This meta-class represents the ability to associate a DiagnosticFunctionIdentifier with a DiagnosticIumpr.	
M2 Parameter	
DiagnosticIumprToFunctionIdentifierMapping	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00736]

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio
BSW Parameter	**BSW Type**
DemIUMPRDenGroup	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter specifies the assigned denominator type	
Template Description	
This meta-class represents the ability to model a IUMPR denominator groups.	
M2 Parameter	
DiagnosticIumprDenominatorGroup	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00838]

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio/DemIUMPRDenGroup
BSW Parameter	**BSW Type**

▽

△

DEM_IUMPR_DEN_500MILL	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Condition based on definition of 500miles conditions as defined for OBD2.	
Template Description	
DiagnosticIumprDenominatorGroup.iumpr: This reference collects DiagnosticIumpr to a DiagnosticIumprDenominatorGroup. **Identifiable.category:** The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.	
M2 Parameter	
DiagnosticIumprDenominatorGroup.iumpr, Identifiable.category	
Mapping Rule	**Mapping Type**
DiagnosticIumprDenominatorGroup.category ==IUMPR_DENOMINATOR_500_MILES	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio/DemIUMPRDenGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_DEN_COLDSTART	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Condition based on definition of "cold start" as defined for EU5+	
Template Description	
DiagnosticIumprDenominatorGroup.iumpr: This reference collects DiagnosticIumpr to a DiagnosticIumprDenominatorGroup. **Identifiable.category:** The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.	
M2 Parameter	
DiagnosticIumprDenominatorGroup.iumpr, Identifiable.category	
Mapping Rule	**Mapping Type**
DiagnosticIumprDenominatorGroup == IUMPR_DENOMINATOR_COLDSTART	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio/DemIUMPRDenGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_DEN_CSERS	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Cold start emission reduction strategy" denominator	
Template Description	
This meta-class represents the ability to model a IUMPR denominator groups.	
M2 Parameter	
DiagnosticIumprDenominatorGroup	
Mapping Rule	**Mapping Type**
DiagnosticIumprDenominatorGroup.category == IUMPR_DENOMINATOR_CSERS	full

▽

△

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemRatio/DemIUMPRDenGroup

BSW Parameter	BSW Type
DEM_IUMPR_DEN_CSERS_API	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
Cold start emission reduction strategy" denominator considering additional conditions reported via Dem_RepIUMPRDen Release.

Template Description
This meta-class represents the ability to model a IUMPR denominator groups.

M2 Parameter
DiagnosticIumprDenominatorGroup

Mapping Rule	Mapping Type
DiagnosticIumprDenominatorGroup.category == IUMPR_DENOMINATOR_CSERS_API	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemRatio/DemIUMPRDenGroup

BSW Parameter	BSW Type
DEM_IUMPR_DEN_EVAP	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
Condition based on definition of "EVAP" conditions as defined for OBD2.

Template Description
DiagnosticIumprDenominatorGroup.iumpr: This reference collects DiagnosticIumpr to a DiagnosticIumprDenominatorGroup. **Identifiable.category:** The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.

M2 Parameter
DiagnosticIumprDenominatorGroup.iumpr, Identifiable.category

Mapping Rule	Mapping Type
DiagnosticIumprDenominatorGroup.category == IUMPR_DENOMINATOR_EVAP	full

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemRatio/DemIUMPRDenGroup

BSW Parameter	BSW Type
DEM_IUMPR_DEN_EVAPPURGEFLOW	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
Condition based on definition of "EVAP purge flow" conditions

Template Description

▽

△

This meta-class represents the ability to model a IUMPR denominator groups.	
M2 Parameter	
DiagnosticIumprDenominatorGroup	
Mapping Rule	**Mapping Type**
DiagnosticIumprDenominatorGroup.category == IUMPR_DENOMINATOR_ EVAPPURGEFLOW	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio/DemIUMPRDenGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_DEN_NONE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
No further condition. Denominator increments based on GENERAL_INDIVIDUAL_DENOMINATOR only.	
Template Description	
DiagnosticIumprDenominatorGroup.iumpr: This reference collects DiagnosticIumpr to a DiagnosticIumprDenominatorGroup. **Identifiable.category:** The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.	
M2 Parameter	
DiagnosticIumprDenominatorGroup.iumpr, Identifiable.category	
Mapping Rule	**Mapping Type**
DiagnosticIumprDenominatorGroup.category == IUMPR_DENOMINATOR_NONE	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio/DemIUMPRDenGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_DEN_PHYS_API	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
GENERAL_INDIVIDUAL_DENOMINATOR condition considering additional physical condition (component activity) computed within the SW-C and reported via Dem_RepIUMPRDenRelease	
Template Description	
DiagnosticIumprDenominatorGroup.iumpr: This reference collects DiagnosticIumpr to a DiagnosticIumprDenominatorGroup. **Identifiable.category:** The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.	
M2 Parameter	
DiagnosticIumprDenominatorGroup.iumpr, Identifiable.category	
Mapping Rule	**Mapping Type**
DiagnosticIumprDenominatorGroup.category == IUMPR_DENOMINATOR_PHYSICAL_API	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemRatio
BSW Parameter	**BSW Type**
DemIUMPRGroup	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter specifies the assigned IUMPR group of the ratio Id. For details see SAE J1979-DA	
Template Description	
DiagnosticIumprGroup.iumprGroupIdentifier: This aggregation allows for the variant modeling of the groupIdentifier. **DiagnosticIumprGroupIdentifier.groupId:** This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.	
M2 Parameter	
DiagnosticIumprGroup.iumprGroupIdentifier, DiagnosticIumprGroupIdentifier.groupId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00737]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_AFRI1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Air Fuel Ratio Imbalance Monitor Bank 1	
Template Description	
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.	
M2 Parameter	
DiagnosticIumprGroupIdentifier.groupId	
Mapping Rule	**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_AFRI1	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_AFRI2	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Air Fuel Ratio Imbalance Monitor Bank 2	
Template Description	
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.	
M2 Parameter	
DiagnosticIumprGroupIdentifier.groupId	
Mapping Rule	**Mapping Type**



△

If groupIdentifier is set to DEM_IUMPR_AFRI2	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_BOOSTPRS	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Boost Pressure Monitor	
Template Description	
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.	
M2 Parameter	
DiagnosticIumprGroupIdentifier.groupId	
Mapping Rule	**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_BOOSTPRS	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_CAT1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Catalyst Monitor - Bank 1	
Template Description	
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.	
M2 Parameter	
DiagnosticIumprGroupIdentifier.groupId	
Mapping Rule	**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_CAT1	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_CAT2	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Catalyst Monitor - Bank 2	
Template Description	

▽

△

This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.	
M2 Parameter	
DiagnosticIumprGroupIdentifier.groupId	
Mapping Rule	**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_CAT2	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_EGR	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
EGR and/or VVT Monitor	
Template Description	
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.	
M2 Parameter	
DiagnosticIumprGroupIdentifier.groupId	
Mapping Rule	**Mapping Type**
if groupIdentifier is set to DEM_IUMPR_EGR	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_EGSENSOR	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Exhaust Gas Sensor Monitor	
Template Description	
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.	
M2 Parameter	
DiagnosticIumprGroupIdentifier.groupId	
Mapping Rule	**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_EGSENSOR.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_EVAP	ECUC-ENUMERATION-LITERAL-DEF

▽

△

BSW Description	
EVAP Monitor	
Template Description	
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.	
M2 Parameter	
DiagnosticIumprGroupIdentifier.groupId	
Mapping Rule	**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_EVAP	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_FLSYS	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Fuel System Monitor	
Template Description	
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.	
M2 Parameter	
DiagnosticIumprGroupIdentifier.groupId	
Mapping Rule	**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_FLSYS	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_NMHCCAT	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
NMHC Catalyst Monitor	
Template Description	
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.	
M2 Parameter	
DiagnosticIumprGroupIdentifier.groupId	
Mapping Rule	**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_NMHCCAT	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup	
BSW Parameter		**BSW Type**
DEM_IUMPR_NOXADSORB		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
NOx Adsorber Monitor		
Template Description		
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.		
M2 Parameter		
DiagnosticIumprGroupIdentifier.groupId		
Mapping Rule		**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_NOXADSORB		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup	
BSW Parameter		**BSW Type**
DEM_IUMPR_NOXCAT		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
NOx/SCR Catalyst Monitor		
Template Description		
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.		
M2 Parameter		
DiagnosticIumprGroupIdentifier.groupId		
Mapping Rule		**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_NOXCAT		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup	
BSW Parameter		**BSW Type**
DEM_IUMPR_OXS1		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
Oxygen Sensor Monitor - Bank 1		
Template Description		
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.		
M2 Parameter		
DiagnosticIumprGroupIdentifier.groupId		
Mapping Rule		**Mapping Type**
If groupIdentifier ist set to DEM_IUMPR_OXS1		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_OXS2	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Oxygen Sensor Monitor - Bank 2	
Template Description	
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.	
M2 Parameter	
DiagnosticIumprGroupIdentifier.groupId	
Mapping Rule	**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_OSX2	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_PF1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Particulate Filter Monitor Bank 1	
Template Description	
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.	
M2 Parameter	
DiagnosticIumprGroupIdentifier.groupId	
Mapping Rule	**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_PF1	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_PF2	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Particulate Filter Monitor Bank 2	
Template Description	
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.	
M2 Parameter	
DiagnosticIumprGroupIdentifier.groupId	
Mapping Rule	**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_PF2	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup	
BSW Parameter		**BSW Type**
DEM_IUMPR_PMFILTER		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
PM Filter Monitor		
Template Description		
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.		
M2 Parameter		
DiagnosticIumprGroupIdentifier.groupId		
Mapping Rule		**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_PMFILTER		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup	
BSW Parameter		**BSW Type**
DEM_IUMPR_PRIVATE		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
Private group (not reported as InfoType $08/$0B)		
Template Description		
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.		
M2 Parameter		
DiagnosticIumprGroupIdentifier.groupId		
Mapping Rule		**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_PRIVATE		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup	
BSW Parameter		**BSW Type**
DEM_IUMPR_SAIR		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
AIR Monitor (Secondary Air)		
Template Description		
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.		
M2 Parameter		
DiagnosticIumprGroupIdentifier.groupId		
Mapping Rule		**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_SAIR		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_SECOXS1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Secondary Oxygen Sensor Monitor - Bank 1	
Template Description	
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.	
M2 Parameter	
DiagnosticIumprGroupIdentifier.groupId	
Mapping Rule	**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_SECOXS1	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemRatio/DemIUMPRGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_SECOXS2	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Secondary Oxygen Sensor Monitor - Bank 2	
Template Description	
This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.	
M2 Parameter	
DiagnosticIumprGroupIdentifier.groupId	
Mapping Rule	**Mapping Type**
If groupIdentifier is set to DEM_IUMPR_SECOXS2	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemRatio
BSW Parameter	**BSW Type**
DemRatioKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines whether the ratio will be calculated API or observer based.	
Template Description	
This attribute controls the behavior of how the ratio is calculated.	
M2 Parameter	
DiagnosticIumpr.ratioKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00741]

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemRatio/DemRatioKind	
BSW Parameter		**BSW Type**
DEM_RATIO_API		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
API based ratio Id		
Template Description		
The calculation is based on the usage of an API.		
M2 Parameter		
DiagnosticIumprKindEnum.apiBased		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemRatio/DemRatioKind	
BSW Parameter		**BSW Type**
DEM_RATIO_OBSERVER		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
Observer based ratio Id		
Template Description		
The calculation is based on the usage of an observer.		
M2 Parameter		
DiagnosticIumprKindEnum.observerBased		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
Dem	Dem/DemGeneral	
BSW Parameter		**BSW Type**
DemResetConfirmedBitOnOverflow		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
This configuration switch defines, whether the confirmed bit is reset or not while an event memory entry will be displaced.		
Template Description		
This attribute defines whether the confirmed bit is reset or not while an event memory entry will be displaced or could not acquire an entry in the event memory.		
M2 Parameter		
DiagnosticCommonProps.resetConfirmedBitOnOverflow		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dem_00799]

BSW Module	BSW Context	
Dem	Dem/DemGeneral	
BSW Parameter		**BSW Type**
DemResetPendingBitOnOverflow		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
This configuration switch defines, whether the pending bit is reset or not while an event memory entry will be displaced.		
Template Description		
This attribute defines whether the pending bit is reset or not while an event memory entry will be displaced or could not acquire an entry in the event memory. In order to be compliant to ISO 14229-1 [17], this parameter needs to be set to "false".		
M2 Parameter		
DiagnosticCommonProps.resetPendingBitOnOverflow		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dem_00991]

BSW Module	BSW Context	
Dem	Dem/DemGeneral	
BSW Parameter		**BSW Type**
DemStatusBitHandlingTestFailedSinceLastClear		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
This configuration switch defines, whether the aging and displacement mechanism shall be applied to the "TestFailedSince LastClear" status bits.		
Template Description		
This attribute defines, whether the aging and displacement mechanism shall be applied to the "TestFailedSinceLastClear" status bits. On the classic platform, the value of this attribute has to be identical for each DiagnosticMemoryDestination.		
M2 Parameter		
DiagnosticMemoryDestination.statusBitHandlingTestFailedSinceLastClear		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dem_00784]

BSW Module	BSW Context	
Dem	Dem/DemGeneral	
BSW Parameter		**BSW Type**
DemStorageCondition		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container contains the configuration (parameters) for storage conditions.		
Template Description		
Specification of a storage condition.		
M2 Parameter		
DiagnosticStorageCondition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_Dem_00728]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemStorageCondition
BSW Parameter	**BSW Type**
DemStorageConditionReplacementEventRef	ECUC-REFERENCE-DEF
BSW Description	
Specifies the reference to an event which is stored to event memory and supports failure analysis.	
Template Description	
Reference to a DiagnosticEvent to which a StorageConditionGroup is assigned.	
M2 Parameter	
DiagnosticEventToStorageConditionGroupMapping.diagnosticEvent	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00893]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemStorageCondition
BSW Parameter	**BSW Type**
DemStorageConditionStatus	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines the initial status for enable or disable of storage of a diagnostic event. The value is the initialization after power up (before this condition is reported the first time). true: storage of a diagnostic event enabled false: storage of a diagnostic event disabled	
Template Description	
Defines the initial status for enable or disable of acceptance/storage of event reports of a diagnostic event. The value is the initialization after power up (before this condition is reported the first time). true: acceptance/storage of a diagnostic event enabled false: acceptance/storage of a diagnostic event disabled	
M2 Parameter	
DiagnosticCondition.initValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00731]

BSW Module	BSW Context
Dem	Dem/DemGeneral
BSW Parameter	**BSW Type**
DemStorageConditionGroup	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration (parameters) for storage condition groups.	

▽

△

Template Description	
Storage condition group which includes one or several storage conditions.	
M2 Parameter	
DiagnosticStorageConditionGroup	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00773]

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemStorageConditionGroup
BSW Parameter	**BSW Type**
DemStorageConditionRef	ECUC-REFERENCE-DEF
BSW Description	
References an enable condition.	
Template Description	
Reference to storageConditions that are part of the StorageConditionGroup.	
M2 Parameter	
DiagnosticStorageConditionGroup.storageCondition	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dem_00768]

G.4 FiM

BSW Module	BSW Context
FiM	FiM/FiMConfigSet
BSW Parameter	**BSW Type**
FiMFID	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container includes symbolic names of all FIDs.	
Template Description	
FunctionInhibitionNeeds: Specifies the abstract needs on the configuration of the Function Inhibition Manager for one Function Identifier (FID). This class currently contains no attributes. Its name can be regarded as a symbol identifying the FID from the viewpoint of the component or module which owns this class. **DiagnosticFunctionIdentifier:** This meta-class represents a diagnostic function identifier (a.k.a. FID).	
M2 Parameter	
FunctionInhibitionNeeds, DiagnosticFunctionIdentifier	
Mapping Rule	**Mapping Type**

▽

△

In case the owner of the FunctionInhibitionNeeds is a BSW module then the FiMFID.shortName= {capitalizedMip}_{ServiceDependency.symbolicNameProps.symbol}.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FiM_00039]

BSW Module	**BSW Context**
FiM	FiM/FiMConfigSet
BSW Parameter	**BSW Type**
FiMInhibitionConfiguration	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container includes all configuration parameters concerning the relationship between event and FID.	
Template Description	
This meta-class represents the ability to define the inhibition of a specific function identifier within the Fim configuration.	
M2 Parameter	
DiagnosticFunctionIdentifierInhibit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FiM_00038]

BSW Module	**BSW Context**
FiM	FiM/FiMConfigSet/FiMInhibitionConfiguration
BSW Parameter	**BSW Type**
FiMInhEventRef	ECUC-REFERENCE-DEF
BSW Description	
Selection of an single DEM Event.	
Template Description	
This represents the alias event applicable for the referencing inhibition source.	
M2 Parameter	
DiagnosticFunctionInhibitSource.event	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FiM_00100]

BSW Module	**BSW Context**
FiM	FiM/FiMConfigSet/FiMInhibitionConfiguration
BSW Parameter	**BSW Type**
FiMInhInhibitionMask	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
The configuration parameter is used to specify the inhibition mask for an event - FID relation.	
Template Description	
This represents the value of the inhibition mask behavior.	
M2 Parameter	

▽

△

DiagnosticFunctionIdentifierInhibit.inhibitionMask	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FiM_00096]

BSW Module	**BSW Context**
FiM	FiM/FiMConfigSet/FiMInhibitionConfiguration/FiMInhInhibitionMask
BSW Parameter	**BSW Type**
FIM_LAST_FAILED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Last Failed - The monitor Status DEM_MONITOR_STATUS_TF is set to 1.	
Template Description	
This represents the inhibition mask behavior "last failed".	
M2 Parameter	
DiagnosticInhibitionMaskEnum.lastFailed	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
FiM	FiM/FiMConfigSet/FiMInhibitionConfiguration/FiMInhInhibitionMask
BSW Parameter	**BSW Type**
FIM_NOT_TESTED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Not Tested this cycle - DEM_MONITOR_STATUS_TNCTOC flag is set.	
Template Description	
This represents the inhibition mask behavior "not tested".	
M2 Parameter	
DiagnosticInhibitionMaskEnum.notTested	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
FiM	FiM/FiMConfigSet/FiMInhibitionConfiguration/FiMInhInhibitionMask
BSW Parameter	**BSW Type**
FIM_TESTED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Tested - DEM_MONITOR_STATUS_TNCTOC flag is not set.	
Template Description	
This represents the inhibition mask behavior "tested".	

▽

△

M2 Parameter	
DiagnosticInhibitionMaskEnum.tested	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
FiM	FiM/FiMConfigSet/FiMInhibitionConfiguration/FiMInhInhibitionMask
BSW Parameter	**BSW Type**
FIM_TESTED_AND_FAILED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Tested and Failed - DEM_MONITOR_STATUS_TF flag is set and DEM_MONITOR_STATUS_TNCTOC flag is not set.	
Template Description	
This represents the inhibition mask behavior "tested and failed".	
M2 Parameter	
DiagnosticInhibitionMaskEnum.testedAndFailed	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
FiM	FiM/FiMConfigSet
BSW Parameter	**BSW Type**
FiMSummaryEvent	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The summarized EventId definition record consists of a summarized event ID and specific Dem Events. This record means that a particular FID that has to be disabled in case of summarized event (defined above) is to be disabled in any of the specific events. A possible solution could be assigning events as summarized events along with a list of specific events. During the configuration process the summarized event substitutes the referenced single events. However, it is not outlined how this requirement is solved - whether by configuration process or by implementation within the FiM. The FiM configuration tool could also build up a suitable data structure for summarized events and deal with it in the FiM implementation.	
Template Description	
This meta-class represents the ability to model a Fim event group, also known as a summary event in Fim terminology. This represents a group of single diagnostic events.	
M2 Parameter	
DiagnosticFimEventGroup	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FiM_00603]

G.5 IdsM

BSW Module	BSW Context
IdsM	IdsM/IdsMConfiguration/IdsMEvent
BSW Parameter	**BSW Type**
IdsMSinkDem	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
The QSEv will be sent to the Dem Module into a Security Event Memory (Sem) to persist it on the local ECU.	
Template Description	
This meta-class represents the ability to map a security event that is defined in the context of the Security Extract to a diagnostic event defined on the context of the DiagnosticExtract.	
M2 Parameter	
DiagnosticEventToSecurityEventMapping	
Mapping Rule	**Mapping Type**
If the (M2) DiagnosticEventToSecurityEventMapping of the DEXT defines a mapping between the (M2) SecurityEventDefinition corresponding to the enclosing IdsMEvent and a (M2) Diagnostic Event, then IdsMSinkDem = TRUE. Otherwise, IdsMSinkDem = FALSE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00035]

G.6 J1939Dcm

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmDspExternalSRDataElementClass/J1939DcmDataElementInstance
BSW Parameter	**BSW Type**
J1939DcmDataElementInstanceRef	ECUC-INSTANCE-REFERENCE-DEF
BSW Description	
Instance Reference to the primitive data which shall be read or written. Supported are VariableDataPrototypes in Sender ReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationPrimitiveDataType of category VALUE or BOOLEAN or if the AutosarDataPrototype is typed with a ImplementationDataType of category VALUE or TYPE_REFERENCE that in turn boils down to VALUE	
Template Description	
This represents the dataElement in the application software that is accessed for diagnostic purpose. This role is applicable on the classic platform.	
M2 Parameter	
DiagnosticServiceDataMapping.mappedDataElement	
Mapping Rule	**Mapping Type**
DiagnosticServiceDataMapping maps to a primitive data.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00067]

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmDspExternalSRDataElementClass/J1939DcmSubElementInDataElementInstance
BSW Parameter	**BSW Type**
J1939DcmSubElementInDataElementInstanceRef	ECUC-INSTANCE-REFERENCE-DEF
BSW Description	
Instance Reference to the primitve sub-element (at any level) of composite data in a port which shall be read. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in Parameter Interfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationCompositeDataType.	
Template Description	
This represents the dataElement in the application software that is accessed for diagnostic purpose. This role is applicable on the classic platform.	
M2 Parameter	
DiagnosticServiceDataMapping.mappedDataElement	
Mapping Rule	**Mapping Type**
DiagnosticServiceDataMapping maps to a primitive element within a composite data, where the AutosarDataPrototype is typed with a ApplicationCompositeDataType.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00066]

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmNode
BSW Parameter	**BSW Type**
J1939DcmServiceOnlyDTCsMemoryDestinationRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the user defined memory used for the Service Only DTCs handled by DM53, DM54, and DM55.	
Template Description	
Reference to the applicable DiagnosticMemoryDestination.	
M2 Parameter	
DiagnosticTroubleCodeProps.diagnosticMemory	
Mapping Rule	**Mapping Type**
This upstream mapping shall only exist for a DiagnosticTroubleCodeJ1939 with attribute kind set to serviceOnly that references the DiagnosticTroubleCodeProps that owns the memoryDestination.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00051]

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions
BSW Parameter	**BSW Type**
J1939DcmModeCondition	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	



△

This container contains the configuration of a mode condition or an environmental conditions which can be used as argument in J1939DcmModeRules. One J1939DcmModeCondition shall contain either one J1939DcmSwcModeRef or one J1939DcmBswModeRef or one J1939DcmSwcSRDataElementRef. Please note that the J1939Dcm acts as well as mode manager. Therefore the references J1939DcmSwcModeRef or one J1939DcmBswModeRef might point to provided ModeDeclarationGroupPrototypes of the J1939Dcm itself as well as to provided ModeDeclarationGroupPrototypes of other Bsw Modules or software components. In case of a configured J1939DcmSwcModeRef or J1939DcmBswModeRef only the J1939DcmConditionType J1939DCM_EQUALS or J1939DCM_EQUALS_NOT are applicable. In case of J1939DcmSwcSRDataElementRef all literals of J1939DcmConditionType are possible.
Template Description
DiagnosticCompareConditions are atomic conditions. They are based on the idea of a comparison at runtime of some variable data with something constant. The type of the comparison (==, !=, <, <=, ...) is specified in DiagnosticCompare Condition.compareType.
M2 Parameter
DiagnosticEnvCompareCondition

Mapping Rule	**Mapping Type**
Depending on the reference a DcmModeCondition is mapped to a DiagnosticEnvModeCondition if only one reference is present and reference is a DcmSwcModeRef or a DcmBswModeRef. If two references are present, a DcmSwcSRDataElementRef and a DcmSwcSRDataElementValueRef, then DcmModeCondition is mapped to a DiagnosticEnvDataCondition.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00071]

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition

BSW Parameter	**BSW Type**
J1939DcmBswModeRef	ECUC-INSTANCE-REFERENCE-DEF
BSW Description	
This parameter references a mode of a ModeDeclarationGroupPrototype provided by a Basic Software Module used for the condition. Please note that such ModeDeclarationGroupPrototype are owned by a Basic Software Module Description in the role providedModeGroup.	
Template Description	
This reference represents both the ModeDeclarationGroupPrototype and the ModeDeclaration relevant for the mode comparison.	
M2 Parameter	
DiagnosticEnvModeCondition.modeElement	

Mapping Rule	**Mapping Type**
For DcmModeRef a new DiagnosticEnvBswModeElement is used, pointing to the ModeDeclaration via ModeInModuleDescriptionInstanceRef. This new DiagnosticEnvModeElement shall be aggregated by the same DiagnosticEnvironmentalConfition as the DiagnosticEnvModeCondition, in which the target of the reference DiagnosticEnvModeCondition.modeElement points to the this DiagnosticEnvModeElement.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00059]

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition
BSW Parameter	**BSW Type**
J1939DcmConditionType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter specifies what kind of comparison that is made for the evaluation of the mode condition.	
Template Description	
This attributes represents the concrete type of the comparison.	
M2 Parameter	
DiagnosticEnvCompareCondition.compareType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00057]

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType
BSW Parameter	**BSW Type**
J1939DCM_EQUALS	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Equal (==)	
Template Description	
equal	
M2 Parameter	
DiagnosticCompareTypeEnum.isEqual	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType
BSW Parameter	**BSW Type**
J1939DCM_EQUALS_NOT	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Not equal (!=)	
Template Description	
not equal	
M2 Parameter	
DiagnosticCompareTypeEnum.isNotEqual	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType	
BSW Parameter	**BSW Type**	
J1939DCM_GREATER_OR_EQUAL	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
Greater or equal (>=)		
Template Description		
greater than or equal		
M2 Parameter		
DiagnosticCompareTypeEnum.isGreaterOrEqual		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType	
BSW Parameter	**BSW Type**	
J1939DCM_GREATER_THAN	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
Greater than (>)		
Template Description		
greater than		
M2 Parameter		
DiagnosticCompareTypeEnum.isGreaterThan		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType	
BSW Parameter	**BSW Type**	
J1939DCM_LESS_OR_EQUAL	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
Less or equal (<=)		
Template Description		
less than or equal		
M2 Parameter		
DiagnosticCompareTypeEnum.isLessOrEqual		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType

BSW Parameter	BSW Type
J1939DCM_LESS_THAN	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
Less than (<)

Template Description
less than

M2 Parameter
DiagnosticCompareTypeEnum.isLessThan

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition

BSW Parameter	BSW Type
J1939DcmSwcModeRef	ECUC-INSTANCE-REFERENCE-DEF

BSW Description
This parameter references a mode in a particular mode request port of a software component that is used for the condition.

Template Description
This reference represents both the ModeDeclarationGroupPrototype and the ModeDeclaration relevant for the mode comparison.

M2 Parameter
DiagnosticEnvModeCondition.modeElement

Mapping Rule	Mapping Type
For DcmModeRef a new DiagnosticEnvSwcModeElement is used, pointing to the ModeDeclaration via PModeInSystemInstanceRef. This new DiagnosticEnvModeElement shall be aggregated by the same DiagnosticEnvironmentalConfition as the DiagnosticEnvModeCondition, in which the target of the reference DiagnosticEnvModeCondition.modeElement points to the this DiagnosticEnvMode Element.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00058]

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition

BSW Parameter	BSW Type
J1939DcmSwcSRDataElementRef	ECUC-REFERENCE-DEF

BSW Description
Reference to environmental conditions. It is possible to reference a S/R Receiver-Port to read physical values and compare (equal, greater, less,...) them with a configured value that is defined by J1939DcmSwcSRDataElementValue.

Template Description
This reference represents the related diagnostic data element.

M2 Parameter
DiagnosticEnvDataCondition.dataElement

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00060]

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmSwcSRDataElementValue/J1939DcmSwcSRDataElementArray/J1939DcmSwcSRDataElementArrayElement

BSW Parameter	BSW Type
J1939DcmSwcSRDataElementArrayElementIndex	ECUC-INTEGER-PARAM-DEF

BSW Description

Index to an array SR data element.

Template Description

DiagnosticEnvDataElementCondition.dataPrototype:
This instanceRef represent the ability to access a data element owned by the application software on the AUTOSAR classic platform.

DataPrototypeInSystemInstanceRef.targetDataPrototype:
This represents the target of the InstanceRef

Ref.index:
This attribute supports the use case to point on specific elements in an array. This is in particular required if arrays are used to implement particular data objects.

The counting of array indices starts with the value 0, i.e. the index of the first array element is 0.

M2 Parameter

`DiagnosticEnvDataElementCondition.dataPrototype`, `DataPrototypeInSystemInstanceRef.targetDataPrototype`, `Ref.index`

Mapping Rule	Mapping Type
DiagnosticDataElement referenced by the DiagnosticEnvDataCondition is an array.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00078]

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmSwcSRDataElementValue/J1939DcmSwcSRDataElementArray/J1939DcmSwcSRDataElementArrayElement

BSW Parameter	BSW Type
J1939DcmSwcSRDataElementArrayElementValue	ECUC-INTEGER-PARAM-DEF

BSW Description

Value of an array SR data element compare value.

Template Description

This aggregation represents the definition of the compare value against which the value taken from the application software shall be compared.

M2 Parameter

`DiagnosticEnvDataElementCondition.compareValue`

Mapping Rule	Mapping Type
DiagnosticDataElement referenced by the DiagnosticEnvDataCondition is an array.	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Dcm_00079]

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmSwcSRDataElementValue/J1939DcmSwcSRDataElementPrimitive
BSW Parameter	**BSW Type**
J1939DcmSwcSRDataElementPrimitiveValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Reference to a primitive SR data element compare value.	
Template Description	
This aggregation represents the definition of the compare value against which the value taken from the application software shall be compared.	
M2 Parameter	
DiagnosticEnvDataElementCondition.compareValue	
Mapping Rule	**Mapping Type**
DiagnosticDataElement referenced by the DiagnosticEnvDataCondition is primitive.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00077]

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions
BSW Parameter	**BSW Type**
J1939DcmModeRule	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration of a mode rule which represents a logical expression with J1939DcmMode Conditions or other J1939DcmModeRules as arguments. All arguments are processed with the operator defined by DcmLogicalOperator, for instance: Argument_A AND Argument_B AND Argument_C	
Template Description	
A DiagnosticEnvConditionFormula embodies the computation instruction that is to be evaluated at runtime to determine if the DiagnosticEnvironmentalCondition is currently present (i.e. the formula is evaluated to true) or not (otherwise). The formula itself consists of parts which are combined by the logical operations specified by DiagnosticEnvConditionFormula.op. If a diagnostic functionality cannot be executed because an environmental condition fails then the diagnostic stack shall send a negative response code (NRC) back to the client. The value of the NRC is directly related to the specific formula and is therefore formalized in the attribute DiagnosticEnvConditionFormula.nrcValue.	
M2 Parameter	
DiagnosticEnvConditionFormula	
Mapping Rule	**Mapping Type**
A DcmModeRule is mapped to a DiagnosticEnvConditionFormula, if this DcmModeRule is a subrule, i.e. it is referenced by a DcmArgumentRef. In addition, a new DiagnosticEnvironmental Condition shall be created with DiagnosticEnvironmentalCondition.formula containing a Diagnostic EnvConditionFormula. In both cases, if no DcmLogicalOperator is present in this DcmModeRule, then DiagnosticEnvConditionFormula shall be set to DiagnosticLogicalOperatorEnum.logicalAnd.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00053]

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeRule
BSW Parameter	**BSW Type**
J1939DcmArgumentRef	ECUC-CHOICE-REFERENCE-DEF
BSW Description	
This is a choice reference either to a mode condition or a an other mode rule serving as sub-expression.	
Template Description	
A DiagnosticEnvConditionFormulaPart can either be a atomic condition, e.g. a DiagnosticEnvCompareCondition, or a DiagnosticEnvConditionFormula, again, which allows arbitrary nesting.	
M2 Parameter	
DiagnosticEnvConditionFormulaPart	
Mapping Rule	**Mapping Type**
Depending on the destination, one DcmArgumentRef is mapped to a DiagnosticEnvCondition Formula if "destination" is a DcmModeRule, and to a DiagnosticEnvCompareCondition, if "destination" is a DcmModeCondition. The order of the aggregation of the DiagnosticEnvCondition FormulaParts shall correspond to the ordering of the index of the DcmArgumentRefs.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00055]

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeRule
BSW Parameter	**BSW Type**
J1939DcmLogicalOperator	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter specifies the logical operator to be used in the logical expression. If the expression only consists of a single condition this parameter shall not be used.	
Template Description	
This attribute represents the concrete operator (supported operators: and, or) of the condition formula.	
M2 Parameter	
DiagnosticEnvConditionFormula.op	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00054]

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeRule/J1939DcmLogicalOperator
BSW Parameter	**BSW Type**
J1939DCM_AND	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
And (&&)	
Template Description	
Logical AND	
M2 Parameter	



△

DiagnosticLogicalOperatorEnum.logicalAnd	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeRule/J1939DcmLogicalOperator
BSW Parameter	**BSW Type**
J1939DCM_OR	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Or (||)	
Template Description	
Logical OR	
M2 Parameter	
DiagnosticLogicalOperatorEnum.logicalOr	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeRule
BSW Parameter	**BSW Type**
J1939DcmModeRuleNrcValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Optional parameter which defines the NRC to be sent in case the mode rule condition is not valid.	
Template Description	
This attribute represents the concrete NRC value that shall be returned if the condition fails.	
M2 Parameter	
DiagnosticEnvConditionFormula.nrcValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00056]

G.7 KeyM

BSW Module	**BSW Context**
KeyM	KeyM/KeyMCertificateElementVerification/KeyMCertificateElementCondition/KeyMCertificateElementConditionValue/KeyMCertificateElementConditionSenderReceiver

▽

△

BSW Parameter	BSW Type
KeyMCertificateElementConditionSenderReceiver	ECUC-INSTANCE-REFERENCE-DEF
BSW Description	
This parameter references a mode in a particular mode request port of a software component that is used for the condition.	
Template Description	
This reference represents both the ModeDeclarationGroupPrototype and the ModeDeclaration relevant for the mode comparison.	
M2 Parameter	
DiagnosticEnvModeCondition.modeElement	
Mapping Rule	**Mapping Type**
For DcmModeRef a new DiagnosticEnvSwcModeElement is used, pointing to the ModeDeclaration via PModeInSystemInstanceRef. This new DiagnosticEnvModeElement shall be aggregated by the same DiagnosticEnvironmentalConfition as the DiagnosticEnvModeCondition, in which the target of the reference DiagnosticEnvModeCondition.modeElement points to the this DiagnosticEnvMode Element.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_KeyM_00052]

H Non-Splitable Elements in the Scope of this Document

This chapter contains a table of all non-splitable model elements in the scope of this document.

For more information about the concept of splitable model elements and how these shall be treated please refer to the [15, TPS Generic Structure Template].

Name of non splitable element
DataPrototypeInSystemInstanceRef.base
DataPrototypeInSystemInstanceRef.contextComponent
DataPrototypeInSystemInstanceRef.contextDataPrototype
DataPrototypeInSystemInstanceRef.contextPort
DataPrototypeInSystemInstanceRef.contextRootComposition
DataPrototypeInSystemInstanceRef.rootDataPrototype
DataPrototypeInSystemInstanceRef.targetDataPrototype
DiagnosticDataIdentifierSet.diagnosticDataIdentifier
DiagnosticEnvConditionFormula.part
DiagnosticEnvDataCondition.compareValue
DiagnosticEnvDataCondition.dataElement
DiagnosticEnvDataElementCondition.compareValue
DiagnosticEnvDataElementCondition.dataPrototype
DiagnosticEnvDataElementCondition.swDataDefProps
DiagnosticEnvModeCondition.modeElement
DiagnosticExtendedDataRecord.element
DiagnosticJ1939ExpandedFreezeFrame.spn
DiagnosticJ1939FreezeFrame.spn
DiagnosticParameter.ident
DiagnosticParameterElementAccess.contextElement
DiagnosticServiceInstance.serviceClass
PModeInSystemInstanceRef.base
PModeInSystemInstanceRef.contextComponent
PModeInSystemInstanceRef.contextComposition
PModeInSystemInstanceRef.contextModeDeclarationGroup
PModeInSystemInstanceRef.contextPPort
PModeInSystemInstanceRef.targetMode
SwcServiceDependencyInSystemInstanceRef.contextRootSwComposition
SwcServiceDependencyInSystemInstanceRef.contextSwComponentPrototype
SwcServiceDependencyInSystemInstanceRef.targetSwcServiceDependency

Table H.1: Usage of non splitable elements

I Variation Points in the Scope of this Document

This chapter contains a table of all model elements stereotyped ≪atpVariation≫ in the scope of this document.

Each entry in the table consists of the identification of the model element itself and the applicable value of the tagged value vh.latestBindingTime.

For more information about the concept of variation points and how model elements that contain variation points shall be treated please refer to the [15, TPS Generic Structure Template].

Variation Point	*Latest Binding Time*
DiagnosticAbstractDataIdentifier.id	preCompileTime
DiagnosticAbstractParameter.dataElement	postBuild
DiagnosticAging.agingCycle	postBuild
DiagnosticAging.threshold	postBuild
DiagnosticCommonProps	codeGenerationTime
DiagnosticCommonProps.debounceAlgorithmProps	postBuild
DiagnosticConnectedIndicator.healingCycleCounterThreshold	postBuild
DiagnosticContributionSet.element	postBuild
DiagnosticContributionSet.serviceTable	postBuild
DiagnosticDataIdentifier.dataElement	postBuild
DiagnosticDataIdentifierSet.diagnosticDataIdentifier	postBuild
DiagnosticDebounceAlgorithmProps.debounceBehavior	preCompileTime
DiagnosticEnableConditionGroup.enableCondition	postBuild
DiagnosticEvent.confirmationThreshold	postBuild
DiagnosticEvent.connectedIndicator	postBuild
DiagnosticEventToOperationCycleMapping.diagnosticOperationCycle	postBuild
DiagnosticFreezeFrame.recordNumber	preCompileTime
DiagnosticIndicator.type	preCompileTime
DiagnosticIumprGroup.iumprGroupIdentifier	postBuild
DiagnosticMeasurementIdentifier.obdMid	preCompileTime
DiagnosticParameterIdentifier.dataElement	postBuild
DiagnosticProtocol.diagnosticConnection	postBuild
DiagnosticProtocol.priority	preCompileTime
DiagnosticProtocol.sendRespPendOnTransToBoot	preCompileTime
DiagnosticProtocol.serviceTable	postBuild
DiagnosticRoutine.id	preCompileTime
DiagnosticServiceTable.diagnosticConnection	postBuild
DiagnosticServiceTable.diagnosticServiceInstance	postBuild
DiagnosticStorageConditionGroup.storageCondition	postBuild
DiagnosticTestIdentifier.id	preCompileTime
DiagnosticTestIdentifier.uasId	preCompileTime
DiagnosticTestResult.diagnosticEvent	preCompileTime
DiagnosticTestResult.updateKind	preCompileTime

▽

△

Variation Point	*Latest Binding Time*
DiagnosticTroubleCodeGroup.dtc	postBuild
DiagnosticTroubleCodeGroup.groupNumber	preCompileTime
DiagnosticTroubleCodeObd.considerPtoStatus	preCompileTime
DiagnosticTroubleCodeObd.eventReadinessGroup	postBuild
DiagnosticTroubleCodeObd.obdDTCValue	preCompileTime
DiagnosticTroubleCodeProps.extendedDataRecord	preCompileTime
DiagnosticTroubleCodeProps.freezeFrame	preCompileTime
DiagnosticTroubleCodeProps.legislatedFreezeFrameContentUdsObd	preCompileTime
DiagnosticTroubleCodeProps.obdRelevance	postBuild
DiagnosticTroubleCodeProps.priority	postBuild
DiagnosticTroubleCodeProps.snapshotRecordContent	preCompileTime
DiagnosticTroubleCodeUds.considerPtoStatus	postBuild
DiagnosticTroubleCodeUds.eventReadinessGroup	postBuild
DiagnosticTroubleCodeUds.functionalUnit	postBuild
DiagnosticTroubleCodeUds.obdDtcValue3Byte	postBuild
DiagnosticTroubleCodeUds.severity	postBuild
DiagnosticTroubleCodeUds.troubleCodeProps	postBuild
DiagnosticTroubleCodeUds.udsDtcValue	postBuild
DiagnosticTroubleCodeUds.wwhObdDtcClass	postBuild
ISignalPortToDiagnosticEventMapping.iSignalTriggering	postBuild

Table I.1: Usage of variation points