

<b>Document Title</b>	Diagnostic Extract Template
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	673

<b>Document Status</b>	Final
<b>Part of AUTOSAR Standard</b>	Classic Platform
<b>Part of Standard Release</b>	4.4.0

<b>Document Change History</b>			
<b>Date</b>	<b>Release</b>	<b>Changed by</b>	<b>Description</b>
2018-10-31	4.4.0	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation</li> </ul>
2017-12-08	4.3.1	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation</li> </ul>
2016-11-30	4.3.0	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Support for OBD</li> <li>• Support for J1939</li> <li>• Support for Fim configuration</li> <li>• Support for environmental conditions</li> <li>• Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation</li> </ul>
2015-07-31	4.2.2	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation</li> </ul>
2014-10-31	4.2.1	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Initial Release</li> </ul>

## 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	9
1.1	Overview	9
1.1.1	OEM	10
1.1.2	Application Developer	11
1.1.3	ECU-Supplier	11
1.1.4	Exchanging of Files	12
1.1.5	Relationship to software-component Service Needs	13
1.1.6	Recommendation and Hints	14
1.1.7	Limitations	14
1.2	Scope	15
1.3	Abbreviations	16
1.4	Document Conventions	18
1.5	Requirements Tracing	21
2	Use Cases	25
2.1	Use cases for diagnostic data exchange	25
2.2	Configuration of DCM	25
2.3	Configuration of DEM	26
2.4	Configuration of the Fim	28
2.4.1	Model Function Inhibition	28
2.4.2	Model Fim configuration before Dem exists	29
2.5	Configuration of J1939 Diagnostics	30
2.5.1	Modeling of J1939 Diagnostic Aspects independent of the Deployment	30
2.5.2	J1939 Diagnostic Content modeled in the Diagnostic Extract	30
3	Conceptual Background	31
3.1	Definition of relevant Diagnostic Elements	31
3.2	Abstraction from EcuC Level	31
3.3	Independence of Definition	32
3.3.1	Use of <code>&lt;&lt;atpSplittable&gt;&gt;</code> enabling separation of elements over several physical files	32
3.3.2	Use of self-contained mapping elements	32
4	Common Meta Model Elements	33
4.1	Introduction	33
4.2	Data Identifier vs. Routine vs. Data Element	33
4.2.1	Usage of <code>SwDataDefProps</code>	40
4.2.2	Definition of Arrays	45
4.2.3	Definition of textual Strings	46
4.3	Textual Documentation	46
4.4	Diagnostic Contribution	50
4.5	Diagnostic Protocol	53
4.6	Diagnostic Common Properties	58

5	Diagnostic Services	63
5.1	Introduction	63
5.2	Service Instance vs. Service Class	63
5.3	Access Permission, Session, Security Level	66
5.3.1	Introduction to Access Permission	66
5.3.2	Prioritization of Access Permission	71
5.4	Environmental Conditions for the Execution of Diagnostic Services	73
5.4.1	Environmental Condition Formula	75
5.4.2	Atomic Conditions	77
5.4.2.1	Data Condition	78
5.4.2.2	Mode Condition	79
5.5	Diagnostic Services supported by AUTOSAR	82
5.5.1	DataByIdentifier	83
5.5.2	IOControl	89
5.5.3	EcuReset	92
5.5.4	ClearDiagnosticInformation	95
5.5.5	Memory Services	96
5.5.6	CommunicationControl	106
5.5.7	DynamicallyDefineDataIdentifier	111
5.5.8	ReadDataByPeriodicIdentifier	114
5.5.9	ControlDTCSetting	117
5.5.10	ResponseOnEvent	118
5.5.11	ReadDTCInformation	123
5.5.12	RoutineControl	125
5.5.13	SecurityAccess	132
5.5.14	SessionControl	134
5.5.15	RequestFileTransfer	136
5.6	OBD Diagnostic Services supported by AUTOSAR	138
5.6.1	OBD Mode 0x01 (RequestCurrentPowertrainDiagnosticData)	141
5.6.2	OBD Mode 0x02 (RequestPowertrainFreezeFrameData)	143
5.6.3	OBD Mode 0x03 / 0x07 (RequestEmissionRelatedDiagnosticTroubleCodes)	145
5.6.4	OBD Mode 0x04 (ClearResetEmissionRelatedDiagnosticInformation)	146
5.6.5	OBD Mode 0x06 (RequestOnBoardMonitoringTestResults)	147
5.6.6	OBD Mode 0x08 (RequestControlOfOnBoardDevice)	149
5.6.7	OBD Mode 0x09 (RequestVehicleInformation)	151
5.6.8	OBD Mode 0x0A (RequestEmissionRelatedDiagnosticTroubleCodesPermanentStatus)	154
5.7	UDS Diagnostic Services for supporting WWH-OBD	156
5.8	Diagnostic Service Mapping	156
5.8.1	Diagnostic Service Data Mapping	160
5.8.2	Diagnostic Service Software Mapping	164
6	Diagnostic Event Handling	168
6.1	Introduction	168

6.2	DiagnosticEvent	168
6.3	DiagnosticTroubleCode	175
6.4	DiagnosticExtendedDataRecord	188
6.5	DiagnosticFreezeFrame	190
6.6	DiagnosticCondition	191
6.7	DiagnosticConditionGroup	193
6.8	DiagnosticMapping	194
6.8.1	DiagnosticEvent to DtcUds Mapping	196
6.8.2	DiagnosticEvent to DiagnosticOperationCycle Mapping	197
6.8.3	DiagnosticEvent to DebounceAlgorithm Mapping	197
6.8.4	DiagnosticEvent to EnableConditionGroup Mapping	204
6.8.5	DiagnosticEvent to StorageConditionGroup Mapping	205
6.8.6	DiagnosticEvent to Port Mapping	206
6.8.7	DiagnosticOperationCycle to Port Mapping	207
6.8.8	DiagnosticEnableCondition to Port Mapping	208
6.8.9	DiagnosticStorageCondition to Port Mapping	209
6.8.10	Provided Data Mapping	210
6.8.11	Master to Slave Event Mapping	212
6.9	DiagnosticOperationCycle	214
6.10	DiagnosticAging	215
6.11	DiagnosticIndicator	216
6.12	DiagnosticTestResult	217
6.13	OBd-related aspects of Dem Configuration	219
6.13.1	Dem Configuration for OBd-II	223
6.13.2	Dem Configuration for WWH-OBd	225
7	Functional Inhibition	226
7.1	Introduction	226
7.2	Alias Events	226
7.3	Function Identifier	227
7.4	Mapping between Inhibition Source and Diagnostic Event	228
7.5	Alias Event Mapping	228
7.6	Mapping of Function Identifiers to the corresponding Monitors	236
8	Diagnostics on J1939	237
8.1	Introduction	237
8.2	Suspect Parameter Number	238
8.3	J1939Dcm-related Modeling	239
8.4	Dem-related Modeling	239
8.5	Mapping between Software-Components and Controller Applications	243
8.6	Mapping between DiagnosticEvent and J1939 DTC	244
A	Mentioned Class Tables	245
B	History of Constraints and Specification Items	271
B.1	Constraint History of this Document according to AUTOSAR R4.2.1	271
B.1.1	Added Specification Items in R4.2.1	271

B.1.2	Added Constraints in R4.2.1 . . . . .	275
B.2	Constraint History of this Document according to AUTOSAR R4.2.2 . . . . .	277
B.2.1	Added Traceables in R4.2.2 . . . . .	277
B.2.2	Changed Traceables in R4.2.2 . . . . .	277
B.2.3	Deleted Traceables in R4.2.2 . . . . .	277
B.2.4	Added Constraints in R4.2.2 . . . . .	277
B.2.5	Changed Constraints in R4.2.2 . . . . .	277
B.2.6	Deleted Constraints in R4.2.2 . . . . .	278
B.3	Constraint History of this Document according to AUTOSAR R4.3.0 . . . . .	278
B.3.1	Added Traceables in R4.3.0 . . . . .	278
B.3.2	Changed Traceables in R4.3.0 . . . . .	280
B.3.3	Deleted Traceables in R4.3.0 . . . . .	280
B.3.4	Added Constraints in R4.3.0 . . . . .	280
B.3.5	Changed Constraints in R4.3.0 . . . . .	281
B.3.6	Deleted Constraints in R4.3.0 . . . . .	281
B.4	Constraint History of this Document according to AUTOSAR R4.3.1 . . . . .	282
B.4.1	Added Traceables in 4.3.1 . . . . .	282
B.4.2	Changed Traceables in 4.3.1 . . . . .	282
B.4.3	Deleted Traceables in 4.3.1 . . . . .	282
B.4.4	Added Constraints in 4.3.1 . . . . .	282
B.4.5	Changed Constraints in 4.3.1 . . . . .	283
B.4.6	Deleted Constraints in 4.3.1 . . . . .	283
B.5	Constraint History of this Document according to AUTOSAR R4.4.0 . . . . .	283
B.5.1	Added Traceables in 4.4.0 . . . . .	283
B.5.2	Changed Traceables in 4.4.0 . . . . .	284
B.5.3	Deleted Traceables in 4.4.0 . . . . .	284
B.5.4	Added Constraints in 4.4.0 . . . . .	284
B.5.5	Changed Constraints in 4.4.0 . . . . .	285
B.5.6	Deleted Constraints in 4.4.0 . . . . .	285
C	Glossary . . . . .	285
D	Modeling of InstanceRef . . . . .	288
D.1	Introduction . . . . .	288
D.2	Modeling . . . . .	289
E	Upstream Mapping . . . . .	294
E.1	Introduction . . . . .	294
E.2	Dcm . . . . .	294
E.3	Dem . . . . .	471
E.4	Fim . . . . .	471
E.5	J1939 Dcm . . . . .	474
F	Splitable Elements in the Scope of this Document . . . . .	486
G	Variation Points in the Scope of this Document . . . . .	487

## References

- [1] Specification of RTE Software  
AUTOSAR\_SWS\_RTE
- [2] Layered Software Architecture  
AUTOSAR\_EXP\_LayeredSoftwareArchitecture
- [3] Specification of ECU Configuration  
AUTOSAR\_TPS\_ECUConfiguration
- [4] ASAM MCD 2D ODX  
<http://www.asam.net>  
ASAM MCD-2D ODX v2.2.0.pdf
- [5] XML Schema Production Rules  
AUTOSAR\_TPS\_XMLSchemaProductionRules
- [6] System Template  
AUTOSAR\_TPS\_SystemTemplate
- [7] Specification of ECU Configuration Parameters (XML)  
AUTOSAR\_MOD\_ECUConfigurationParameters
- [8] Software Component Template  
AUTOSAR\_TPS\_SoftwareComponentTemplate
- [9] Basic Software Module Description Template  
AUTOSAR\_TPS\_BSWModuleDescriptionTemplate
- [10] Specification of Diagnostic Communication Manager  
AUTOSAR\_SWS\_DiagnosticCommunicationManager
- [11] Specification of Diagnostic Event Manager  
AUTOSAR\_SWS\_DiagnosticEventManager
- [12] Standardization Template  
AUTOSAR\_TPS\_StandardizationTemplate
- [13] Requirements on Diagnostic Extract Template  
AUTOSAR\_RS\_DiagnosticExtractTemplate
- [14] Specification of Function Inhibition Manager  
AUTOSAR\_SWS\_FunctionInhibitionManager
- [15] SAE J1939 Top Level Document
- [16] Unified diagnostic services (UDS) – Part 1: Specification and requirements (Release 2006-12)  
<http://www.iso.org>
- [17] Road vehicles – End-of-life activation of on-board pyrotechnic devices – Part 2: Communication requirements

<http://www.iso.org>

- [18] Road vehicles – Communication between vehicle and external equipment for emission-related diagnostic – Part 5: Emission-related diagnostic services.

<http://www.iso.org>

- [19] Road vehicles – Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements – Part 3: Common message dictionary

<http://www.iso.org>

- [20] SAE J1939-73 Application Layer – Diagnostics

- [21] Software Process Engineering Meta-Model Specification

<http://www.omg.org/spec/SPEM/2.0/>

- [22] Generic Structure Template

AUTOSAR\_TPS\_GenericStructureTemplate



# 1 Introduction

## 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, all SW-Cs above the RTE [1] (Application Software) introduce ports to be connected to the BSW modules [2]. Additionally, SW-Cs express needs which shall be fulfilled by the BSW.

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

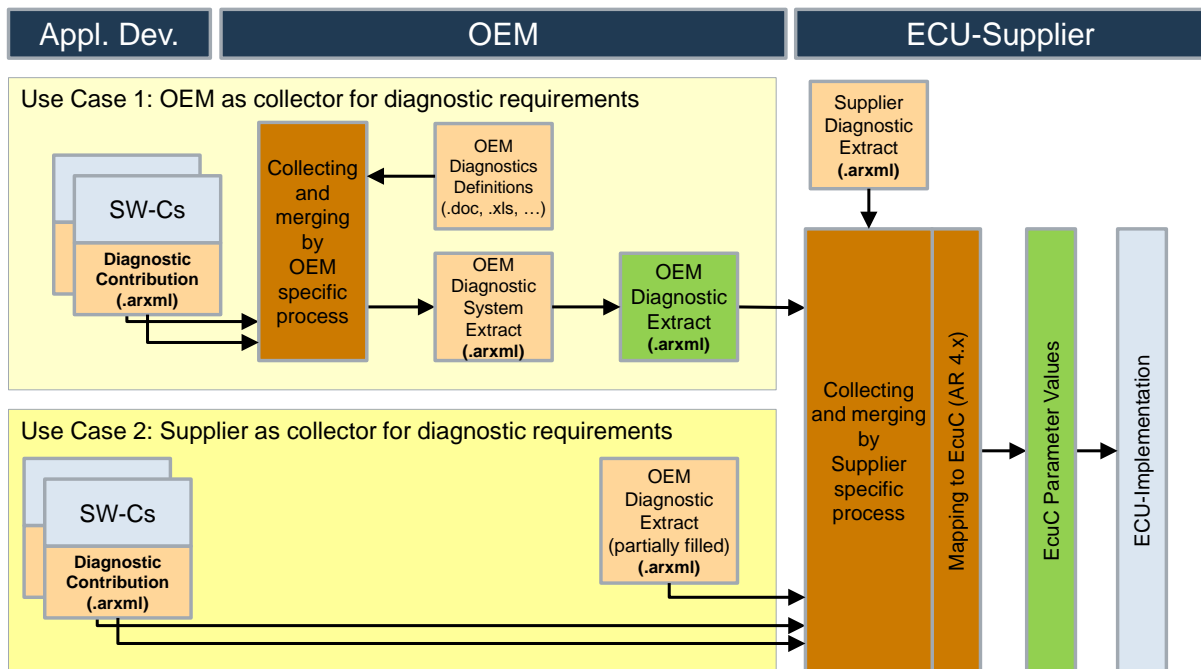
In the past, due to the absence of integral options, many different file formats like ODX or EcuC [3] are often used. But neither ODX nor EcuC is well suited to transfer this information.

For example, ODX [4] 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 new standardized AUTOSAR exchange format on diagnostic functionality that can be used similar to a System Description, formalized as an ARXML [5] file.

In this spirit, the configuration of diagnostic functionality becomes similar to the configuration of the communication part within the System Description [6].



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

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 configuration (M2 to M1 mapping, see also [3] and [7]).

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 configuration 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:

- Definition of the content of a specific `ReadDataByIdentifier` implemented by this software-component
- Definition of the events needed for this software-component

**NOTE:** Only as example, 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 **OEMs** 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 In-

**tegrator** is to integrate all delivered [DiagnosticExtract](#) and to generate the EcuC configuration 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 SW 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.

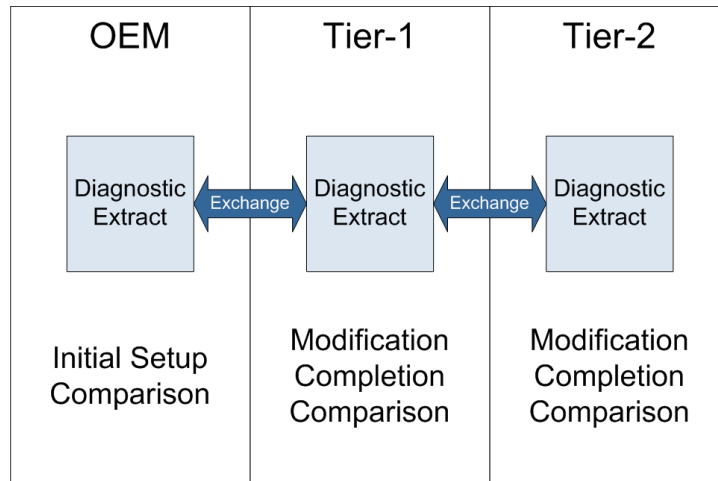
#### 1.1.4 Exchanging of Files

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.

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.



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

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 BSW module DCM or DEM in order to provide the corresponding configuration on BSW level.

The usage of Service Needs is only possible on Atomic software-component level whereas the assignment of diagnostic demands must be possible on Composition level.

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**.

In case the **OEM** does not have his 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** 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 SW inclusive their corresponding `DiagnosticExtract`.

### 1.1.7 Limitations

This first release of the `DiagnosticExtract` template focuses on defining diagnostic requirements a single ECU only. That means that currently, 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 is expected to be extended to also cover configuration of distributed diagnostic functionality. Similar to the description of communication dependencies in the System Description, it shall be possible to describe diagnostic demands on system level to derive the diagnostic demands for a specific ECU from this description.

In general, the `DiagnosticExtract` does not support process-related parts to document the maturity of diagnostic configuration data. This means, that a data object

cannot be marked as “draft” or “released”. This issue needs to be solved by AUTOSAR within a general concept in a future release. Therefore, it does not make sense to introduce a solution for diagnostics only.

## 1.2 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 Software Component Template [8] or Basic Software ModuleDescription Template [9].

Further relations exist to the specification of communication in AUTOSAR systems as described by the System Template [6].

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

The relation of the [DiagnosticExtract](#) to the rest of the AUTOSAR meta-model is sketched in Figure 1.3.

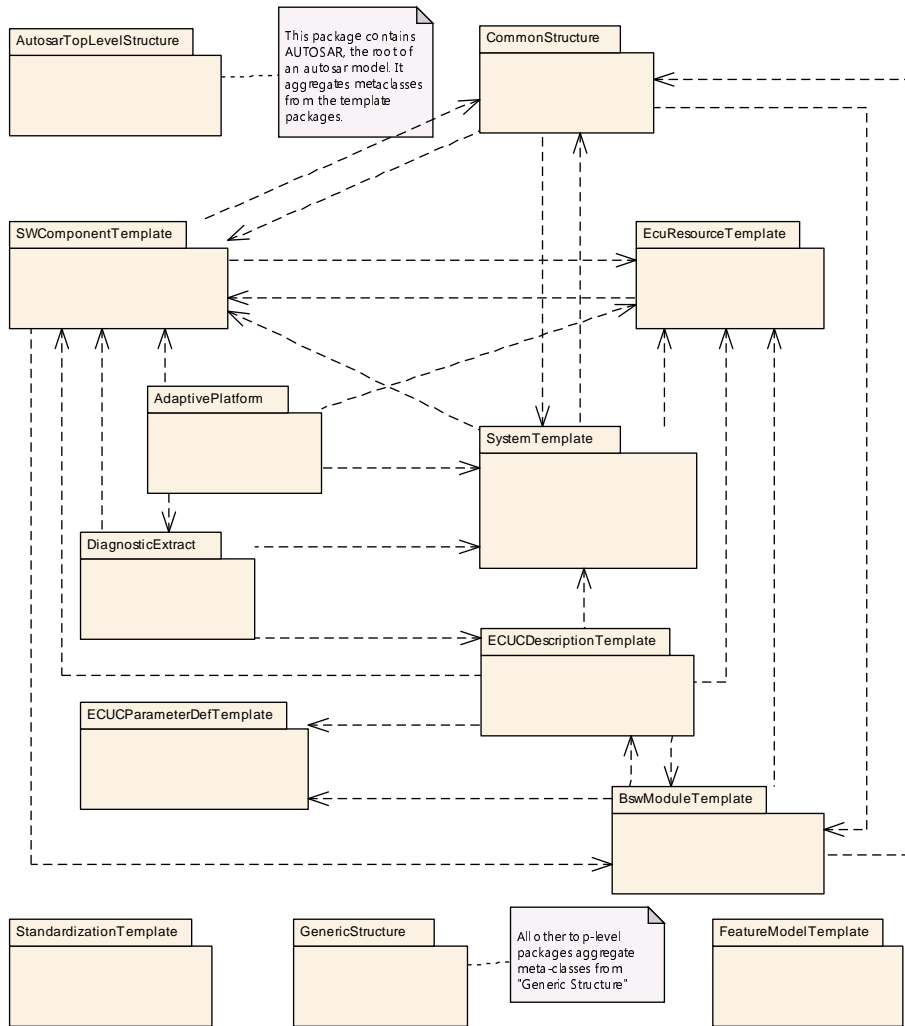


Figure 1.3: The relation of the **DiagnosticExtract** to the rest of the AUTOSAR meta-model

### 1.3 Abbreviations

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
BSW	Basic Software
BswM	Basic Software Manager
CAN	Controller Area Network
CSE	Codes for Scaling Units







<i>Abbreviation</i>	<i>meaning</i>
DEM	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
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
TX	Transmit
SWS	Software Specification
UDS	Unified Diagnostic Services
UML	Unified Modeling Language





Abbreviation	meaning
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 1.1: Abbreviations used in the scope of this Document**

## 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. `PortPrototypes`. 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:

<b>Class</b>	<b>AUTOSAR</b>			
<b>Package</b>	M2::AUTOSARTemplates::AutosarTopLevelStructure			
<b>Note</b>	Root element of an AUTOSAR description, also the root element in corresponding XML documents. <b>Tags:</b> xml.globalElement=true			
<b>Base</b>	ARObject			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
adminData	AdminData	0..1	aggr	This represents the administrative data of an Autosar file. <b>Tags:</b> xml.sequenceOffset=10
arPackage	<a href="#">ARPackage</a>	*	aggr	This is the top level package in an AUTOSAR model. <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30





Class	AUTOSAR			
fileInfo Comment	FileInfoComment	0..1	aggr	This represents a possibility to provide a structured comment in an AUTOSAR file.  <b>Stereotypes:</b> atpStructuredComment <b>Tags:</b> xml.roleElement=true xml.sequenceOffset=-10 xml.typeElement=false
introduction	<a href="#">DocumentationBlock</a>	0..1	aggr	This represents an introduction on the Autosar file. It is intended for example to represent disclaimers and legal notes.  <b>Tags:</b> xml.sequenceOffset=20

**Table 1.2: AUTOSAR**

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 Standardization Template, chapter Support for Traceability ([12]).

The representation of requirements in AUTOSAR documents follows the table specified in [TPS\_STDT\_00078], see Standardization Template, chapter Support for Traceability ([12]).

## 1.5 Requirements Tracing

The following table references the requirements specified in [13] and links to the fulfillment of these.

Requirement	Description	Satisfied by
[RS_DEXT_00001]	Diagnostic data exchange	[TPS_DEXT_01003] [TPS_DEXT_01004] [TPS_DEXT_01005] [TPS_DEXT_01007] [TPS_DEXT_01008] [TPS_DEXT_01014] [TPS_DEXT_01019] [TPS_DEXT_01020] [TPS_DEXT_01022] [TPS_DEXT_01023] [TPS_DEXT_01024] [TPS_DEXT_01025] [TPS_DEXT_01026] [TPS_DEXT_01027] [TPS_DEXT_01028] [TPS_DEXT_01029] [TPS_DEXT_01046] [TPS_DEXT_01055] [TPS_DEXT_01056] [TPS_DEXT_01057] [TPS_DEXT_01060] [TPS_DEXT_01062] [TPS_DEXT_01063] [TPS_DEXT_01066] [TPS_DEXT_01069] [TPS_DEXT_01075]
[RS_DEXT_00002]	No description	[TPS_DEXT_01003] [TPS_DEXT_01004] [TPS_DEXT_01005] [TPS_DEXT_01038] [TPS_DEXT_01047] [TPS_DEXT_01055] [TPS_DEXT_01062] [TPS_DEXT_01063]
[RS_DEXT_00003]	SessionControl	[TPS_DEXT_01039] [TPS_DEXT_01045] [TPS_DEXT_01081] [TPS_DEXT_01082]
[RS_DEXT_00004]	ECUReset	[TPS_DEXT_01019] [TPS_DEXT_01020] [TPS_DEXT_01021] [TPS_DEXT_01045] [TPS_DEXT_01056]
[RS_DEXT_00005]	ClearDiagnosticInformation	[TPS_DEXT_01022] [TPS_DEXT_01045]
[RS_DEXT_00006]	ReadDTCInformation	[TPS_DEXT_01034] [TPS_DEXT_01045] [TPS_DEXT_01060]
[RS_DEXT_00007]	ReadDataByIdentifier	[TPS_DEXT_01045] [TPS_DEXT_01050] [TPS_DEXT_01054] [TPS_DEXT_01146]
[RS_DEXT_00008]	ReadMemoryByAddress	[TPS_DEXT_01024] [TPS_DEXT_01045]
[RS_DEXT_00009]	SecurityAccess	[TPS_DEXT_01036] [TPS_DEXT_01037] [TPS_DEXT_01038] [TPS_DEXT_01045] [TPS_DEXT_01053]
[RS_DEXT_00010]	CommunicationControl	[TPS_DEXT_01029] [TPS_DEXT_01030] [TPS_DEXT_01031] [TPS_DEXT_01032] [TPS_DEXT_01045] [TPS_DEXT_01057] [TPS_DEXT_01074]
[RS_DEXT_00011]	ReadDataByPeriodicIdentifier	[TPS_DEXT_01045]
[RS_DEXT_00012]	DynamicallyDefineDataIdentifier	[TPS_DEXT_01045]
[RS_DEXT_00013]	WriteDataByIdentifier	[TPS_DEXT_01045] [TPS_DEXT_01050] [TPS_DEXT_01054]
[RS_DEXT_00014]	IOControl	[TPS_DEXT_01015] [TPS_DEXT_01016] [TPS_DEXT_01017] [TPS_DEXT_01018] [TPS_DEXT_01045] [TPS_DEXT_01051]
[RS_DEXT_00015]	RoutineControl	[TPS_DEXT_01035] [TPS_DEXT_01045] [TPS_DEXT_01049] [TPS_DEXT_01077] [TPS_DEXT_01078] [TPS_DEXT_01079] [TPS_DEXT_01080]
[RS_DEXT_00016]	RequestDownload	[TPS_DEXT_01027] [TPS_DEXT_01045]
[RS_DEXT_00017]	RequestUpload	[TPS_DEXT_01028] [TPS_DEXT_01045]
[RS_DEXT_00018]	TransferData	[TPS_DEXT_01026] [TPS_DEXT_01045]

[RS_DEXT_00019]	RequestTransferExit	[TPS_DEXT_01025]	[TPS_DEXT_01045]
[RS_DEXT_00020]	WriteMemoryByAddress	[TPS_DEXT_01023]	[TPS_DEXT_01045]
[RS_DEXT_00021]	ControlDTCSetting	[TPS_DEXT_01045]	[TPS_DEXT_01075] [TPS_DEXT_01076]
[RS_DEXT_00022]	ResponseOnEvent	[TPS_DEXT_01033]	[TPS_DEXT_01045]
[RS_DEXT_00023]	Configuration of events	[TPS_DEXT_01048]	[TPS_DEXT_01067] [TPS_DEXT_01068] [TPS_DEXT_01069] [TPS_DEXT_01083] [TPS_DEXT_01084] [TPS_DEXT_01085] [TPS_DEXT_03002] [TPS_DEXT_03003] [TPS_DEXT_03004] [TPS_DEXT_03005] [TPS_DEXT_03007] [TPS_DEXT_03011] [TPS_DEXT_03015] [TPS_DEXT_03016]
[RS_DEXT_00024]	Configuration of DTCs	[TPS_DEXT_01064]	[TPS_DEXT_01065] [TPS_DEXT_01066] [TPS_DEXT_01086] [TPS_DEXT_03000] [TPS_DEXT_03003] [TPS_DEXT_03012] [TPS_DEXT_03013] [TPS_DEXT_03014]
[RS_DEXT_00025]	Combined Events	[TPS_DEXT_03003]	
[RS_DEXT_00026]	Enable Conditions	[TPS_DEXT_03015]	[TPS_DEXT_03018]
[RS_DEXT_00027]	Storage Conditions	[TPS_DEXT_03001]	[TPS_DEXT_03006] [TPS_DEXT_03010] [TPS_DEXT_03016] [TPS_DEXT_03019]
[RS_DEXT_00028]	Enable Condition Groups	[TPS_DEXT_01084]	[TPS_DEXT_03010] [TPS_DEXT_03015]
[RS_DEXT_00029]	Storage Condition Groups	[TPS_DEXT_01084]	[TPS_DEXT_03016]
[RS_DEXT_00030]	Assignment of Enable Condition Groups	[TPS_DEXT_03010]	
[RS_DEXT_00031]	Assignment of Storage Condition Group	[TPS_DEXT_03010]	
[RS_DEXT_00032]	Configuration of Extended Data Records	[TPS_DEXT_03008]	
[RS_DEXT_00033]	Configuration of Snapshot Records	[TPS_DEXT_01143]	[TPS_DEXT_01144] [TPS_DEXT_03009]
[RS_DEXT_00034]	Description of Data Identifiers	[TPS_DEXT_01000]	[TPS_DEXT_01001] [TPS_DEXT_01002] [TPS_DEXT_01017] [TPS_DEXT_01050] [TPS_DEXT_01054] [TPS_DEXT_01072] [TPS_DEXT_01134] [TPS_DEXT_01135] [TPS_DEXT_01136] [TPS_DEXT_01138] [TPS_DEXT_01146]
[RS_DEXT_00035]	Description of Dynamic Data Identifiers	[TPS_DEXT_01000]	
[RS_DEXT_00036]	Description of Routine Identifiers	[TPS_DEXT_01088]	
[RS_DEXT_00037]	Description of I/O Identifiers	[TPS_DEXT_01089]	
[RS_DEXT_00038]	Description of array data types	[TPS_DEXT_01001]	[TPS_DEXT_01002]
[RS_DEXT_00039]	Diagnostic Service Table	[TPS_DEXT_01006]	
[RS_DEXT_00040]	Diagnostic Sessions	[TPS_DEXT_01011]	[TPS_DEXT_01081] [TPS_DEXT_01082] [TPS_DEXT_01139]
[RS_DEXT_00041]	Access Permissions	[TPS_DEXT_01012]	[TPS_DEXT_01052] [TPS_DEXT_01061] [TPS_DEXT_01062] [TPS_DEXT_01063] [TPS_DEXT_01071]
[RS_DEXT_00042]	Security Levels	[TPS_DEXT_01012]	[TPS_DEXT_01038] [TPS_DEXT_01053]

[RS_DEXT_00043]	Description of data elements	[TPS_DEXT_01142]	[TPS_DEXT_03020]
[RS_DEXT_00045]	Textual descriptions	[TPS_DEXT_01064] [TPS_DEXT_01066] [TPS_DEXT_01068] [TPS_DEXT_01071]	[TPS_DEXT_01065] [TPS_DEXT_01067] [TPS_DEXT_01069]
[RS_DEXT_00047]	Custom Diagnostic Service	[TPS_DEXT_01009] [TPS_DEXT_01021] [TPS_DEXT_01031]	[TPS_DEXT_01010] [TPS_DEXT_01030]
[RS_DEXT_00049]	Properties of individual diagnostic services	[TPS_DEXT_01013] [TPS_DEXT_01061]	[TPS_DEXT_01052]
[RS_DEXT_00050]	Properties of all diagnostic services of a given kind	[TPS_DEXT_01061]	
[RS_DEXT_00051]	Subfunctions of Diagnostic Services	[TPS_DEXT_01013] [TPS_DEXT_01018] [TPS_DEXT_01020] [TPS_DEXT_01022] [TPS_DEXT_01024] [TPS_DEXT_01026] [TPS_DEXT_01028] [TPS_DEXT_01030] [TPS_DEXT_01034] [TPS_DEXT_01056] [TPS_DEXT_01060] [TPS_DEXT_01076]	[TPS_DEXT_01014] [TPS_DEXT_01019] [TPS_DEXT_01021] [TPS_DEXT_01023] [TPS_DEXT_01025] [TPS_DEXT_01027] [TPS_DEXT_01029] [TPS_DEXT_01031] [TPS_DEXT_01039] [TPS_DEXT_01057] [TPS_DEXT_01075] [TPS_DEXT_01078]
[RS_DEXT_00052]	Mapping of diagnostic services to the PortPrototypes of ApplicationSwComponentTypes	[TPS_DEXT_01040] [TPS_DEXT_01042] [TPS_DEXT_01044] [TPS_DEXT_01050] [TPS_DEXT_01142] [TPS_DEXT_03007] [TPS_DEXT_03018] [TPS_DEXT_03020]	[TPS_DEXT_01041] [TPS_DEXT_01043] [TPS_DEXT_01049] [TPS_DEXT_01051] [TPS_DEXT_03002] [TPS_DEXT_03017] [TPS_DEXT_03019]
[RS_DEXT_00053]	Debouncing of diagnostic events	[TPS_DEXT_01048] [TPS_DEXT_03005]	[TPS_DEXT_03004] [TPS_DEXT_03017]
[RS_DEXT_00054]	Operation cycles	[TPS_DEXT_01086]	[TPS_DEXT_01087]
[RS_DEXT_00055]	Aging	[TPS_DEXT_03021]	
[RS_DEXT_00056]	Indicator	[TPS_DEXT_03022]	
[RS_DEXT_00057]	RequestFileTransfer	[TPS_DEXT_01090]	
[RS_DEXT_00058]	Indicate that an ECU supports ODB	[TPS_DEXT_01122]	
[RS_DEXT_00059]	Support for different protocols	[TPS_DEXT_01124]	
[RS_DEXT_00060]	Function	[TPS_DEXT_01096] [TPS_DEXT_01098] [TPS_DEXT_01100] [TPS_DEXT_01121]	[TPS_DEXT_01097] [TPS_DEXT_01099] [TPS_DEXT_01101]
[RS_DEXT_00061]	Relationship between functions and diagnostic events	[TPS_DEXT_01095] [TPS_DEXT_01099] [TPS_DEXT_01101]	[TPS_DEXT_01098] [TPS_DEXT_01100]
[RS_DEXT_00062]	Pre-configuration of the Fim when the Dem configuration is not yet available	[TPS_DEXT_01095]	
[RS_DEXT_00063]	Relation between functions on Fim level and software-components	[TPS_DEXT_01102]	

[RS_DEXT_00064]	Definition of an SPN	[TPS_DEXT_01103] [TPS_DEXT_01106]
[RS_DEXT_00065]	Definition of freeze frames on J1939	[TPS_DEXT_01104] [TPS_DEXT_01105]
[RS_DEXT_00066]	Mapping between a J1939 controller application and a software-component	[TPS_DEXT_01108]
[RS_DEXT_00067]	Definition of J1939 DTC	[TPS_DEXT_01107] [TPS_DEXT_01145]
[RS_DEXT_00068]	Definition of a Diagnostic Parameter Identifier	[TPS_DEXT_01092]
[RS_DEXT_00069]	Support for OBD Mode 0x01 (RequestCurrentPowertrainDiagnosticData)	[TPS_DEXT_01125]
[RS_DEXT_00070]	Support for OBD Mode 0x02 (RequestPowertrainFreezeFrameData)	[TPS_DEXT_01126]
[RS_DEXT_00071]	Support for OBD ModeModes 0x03 / 0x07 / 0x0A (RequestEmissionRelatedDiagnosticTroubleCodes)	[TPS_DEXT_01127]
[RS_DEXT_00072]	Support for OBD Mode 0x04 (ClearResetEmissionRelatedDiagnosticInformation)	[TPS_DEXT_01128]
[RS_DEXT_00073]	Support for OBD Mode 0x06 (RequestOnBoardMonitoringTestResults)	[TPS_DEXT_01129] [TPS_DEXT_01141]
[RS_DEXT_00074]	Support for OBD Mode 0x08 (RequestControlOfOnBoardDevice)	[TPS_DEXT_01130]
[RS_DEXT_00075]	Support for OBD Mode 0x09 (RequestVehicleInformation)	[TPS_DEXT_01131]
[RS_DEXT_00076]	Definition of Diagnostic Test Identifier	[TPS_DEXT_01132]
[RS_DEXT_00077]	Description of the utilization of UDS for supporting WWH-OBDD	[TPS_DEXT_01133]
[RS_DEXT_00078]	Support for In Use Monitor Performance Ratio	[TPS_DEXT_01110]
[RS_DEXT_00079]	Support for environment conditions	[TPS_DEXT_01113] [TPS_DEXT_01114] [TPS_DEXT_01115] [TPS_DEXT_01116] [TPS_DEXT_01117] [TPS_DEXT_01118] [TPS_DEXT_01119] [TPS_DEXT_01120]

**Table 1.3: RequirementsTracing**



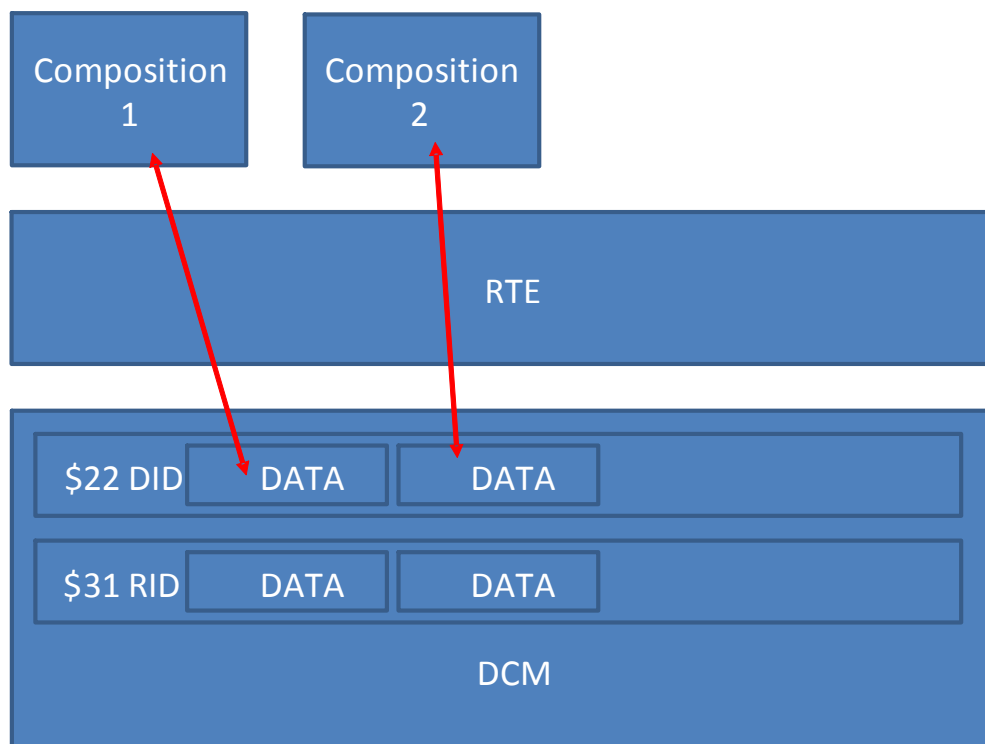
## 2 Use Cases

### 2.1 Use cases for diagnostic data exchange

The basic usage of the `DiagnosticExtract` is the exchange of diagnostic data between the different parties involved in the diagnostic development process to allow the configuration of the DCM and the DEM and to provide the description of corresponding application interfaces to implement diagnostic services and fault handling.

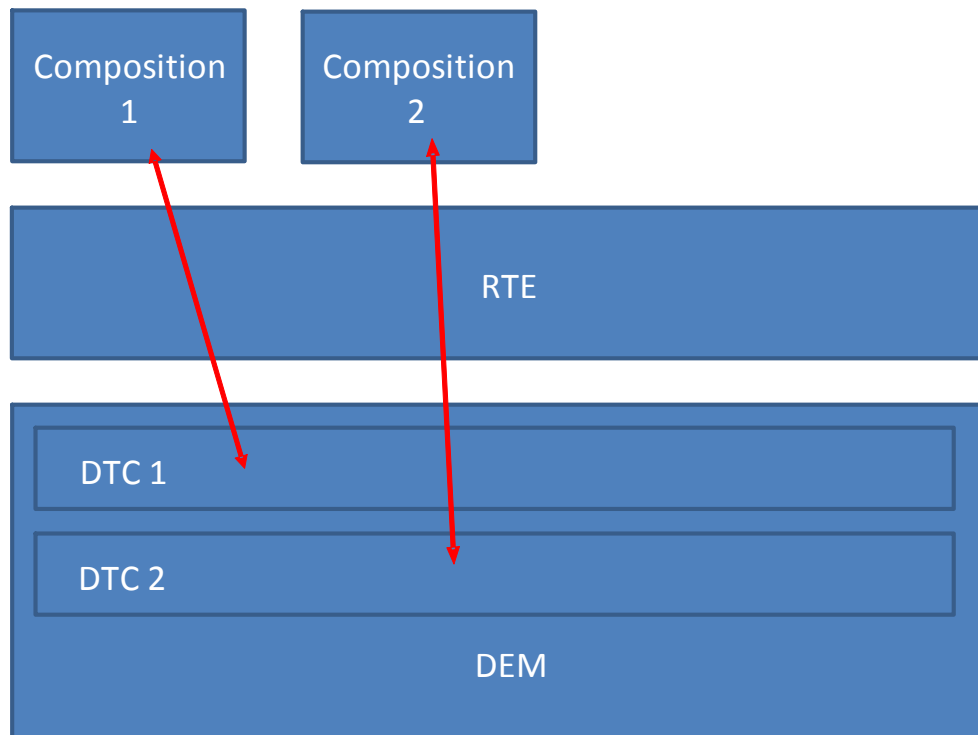
### 2.2 Configuration of DCM

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



**Figure 2.2: Assignment of DEM-related data objects**

The configuration of the DEM includes fault memory data (DTCs and environmental data) and the assignment of corresponding data by 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 side, 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 of completely or partly-filled [DiagnosticExtract](#) to OEM for:
  - Documentation

- ECU testing
- Integration reviews
- Failure correction

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
- Configure 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)

Refinement of use cases:

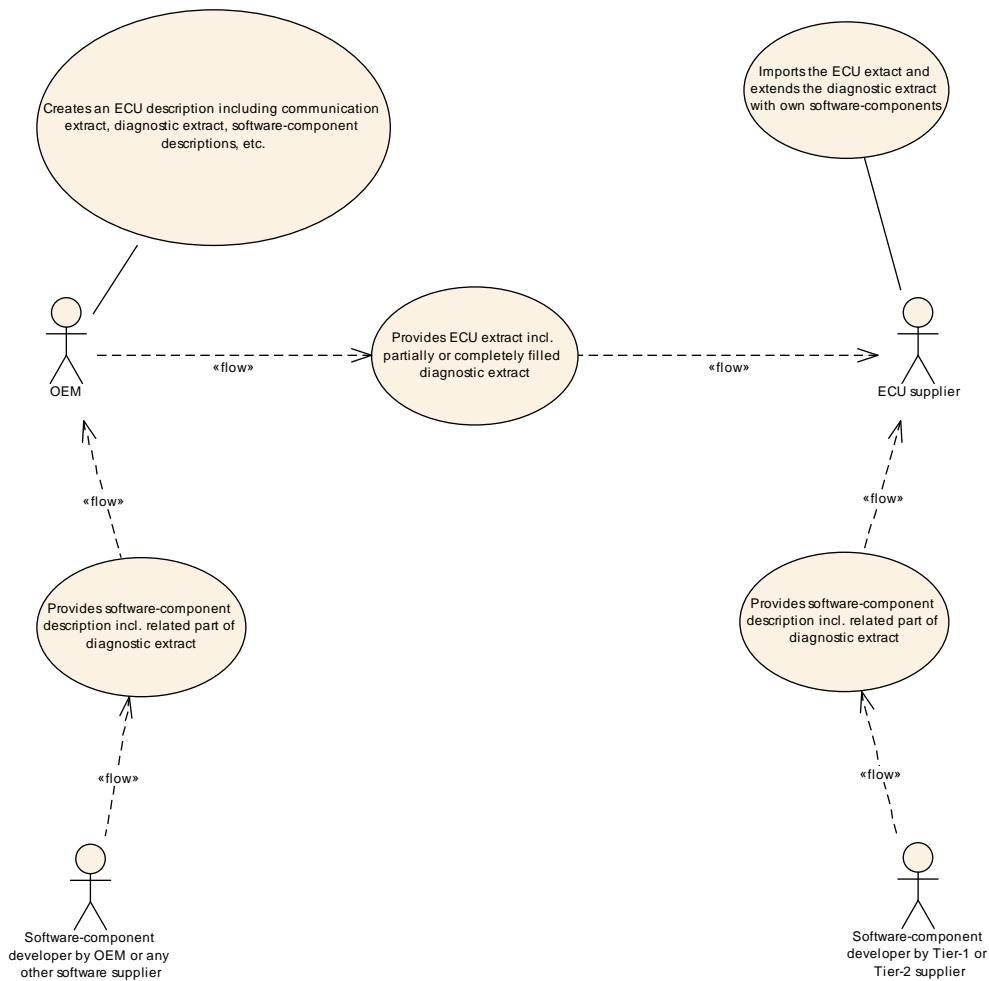
The OEM already provides a System Template for an ECU which describes the ECU Supplier SW parts as a `CompositionSwComponentType` where only inputs and outputs are known.

1. The OEM creates a `DiagnosticExtract` which describes the diagnostic interfaces of an ECU. Supported services are described (e.g. RDBI/WDBI/Routine Control) as well as their corresponding input/output parameters and return values (those which are optional in UDS standard).
2. SWC Developer at OEM or OEM SW Supplier develops SWC and also describes the Diagnostic information using `ServiceNeeds` as diagnostic contribution.
3. The OEM Diagnose Responsible for a Project creates the mappings between the `DiagnosticExtract` and the SWC available on 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 receives the ECU Extract including `DiagnosticExtract` from the OEM and imports it to the project.

6. In the same way is in point 2., the SWC Developer on supplier or Tier 2 side describes the Diagnostic information using *ServiceNeeds* as diagnostic contribution.

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.

For the usage of indicators, it might happen that indicators defined on BSW level in DEM might not be automatically mapped to the implementation on SWC level. This would then require a manual mapping step by the integrator to resolve the mismatch.



**Figure 2.3: Workflow scenario for diagnostic data exchange**

## 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” [14]. 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 is contradicting 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 from each other because the work product of one is the immediate prerequisite that the other organization can start working.

Therefore one important use case for the `DiagnosticExtract` is the ability to configure the Fim module in the scope of the `DiagnosticExtract` **without the immediate need to already have 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 the *diagnostic events* contained in the actual Dem configuration.

## 2.5 Configuration of J1939 Diagnostics

Within the System Extract, J1939 diagnostic is configured on a service level by the diagnostic messages DM01 .. DM57, which will be 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 SAE J1939 Standard [15]) 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 the J1939 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 where 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 valued 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 on 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.

### 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. parameter on buffer sizes). ]([RS\\_DEXT\\_00001](#))

**[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. ]([RS\\_DEXT\\_00002](#))

Therefore, the identification of all diagnostic properties and requirements as 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 possible with EcuC parameters. The approach of “decentralized configuration” is met in the `DiagnosticExtract` template in mainly two ways described in the following sub-chapters.

#### 3.3.1 Use of `<<atpSplittable>>` 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 involved companies respectively roles.

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 any time after the diagnostic elements to be mapped together have been defined. Alternatively, a mapping element can be created after only one diagnostic element has been defined indicating the need to make the mapping complete by defining the additional diagnostic element(s) to map to.



## 4 Common Meta Model Elements

### 4.1 Introduction

This chapter contains a description of the meta-model for the specification of the `DiagnosticExtract` in AUTOSAR. The goal of the specification of the `DiagnosticExtract` is to contribute to the description of the configuration of the `Diagnostic Communication Manager` [10] (Dcm) and the `Diagnostic Event Manager` [11] (Dem)

The meta-model can be roughly divided into five sections

- A common section that contains meta-classes shared between the description of the diagnostic services (that roughly corresponds to the Dcm) and the diagnostic event handling (that roughly corresponds to the Dem), see section 4.2.
- A section dedicated to the configuration of the diagnostic services, see section 5.
- A section dedicated to the configuration of the diagnostic event handling, see section 6.
- A section dedicated to the configuration of the functional inhibition from the diagnostics point of view, see section 7.
- A section dedicated to the configuration of the diagnostics on J1939, see section 8.

### 4.2 Data Identifier vs. Routine vs. Data Element

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.

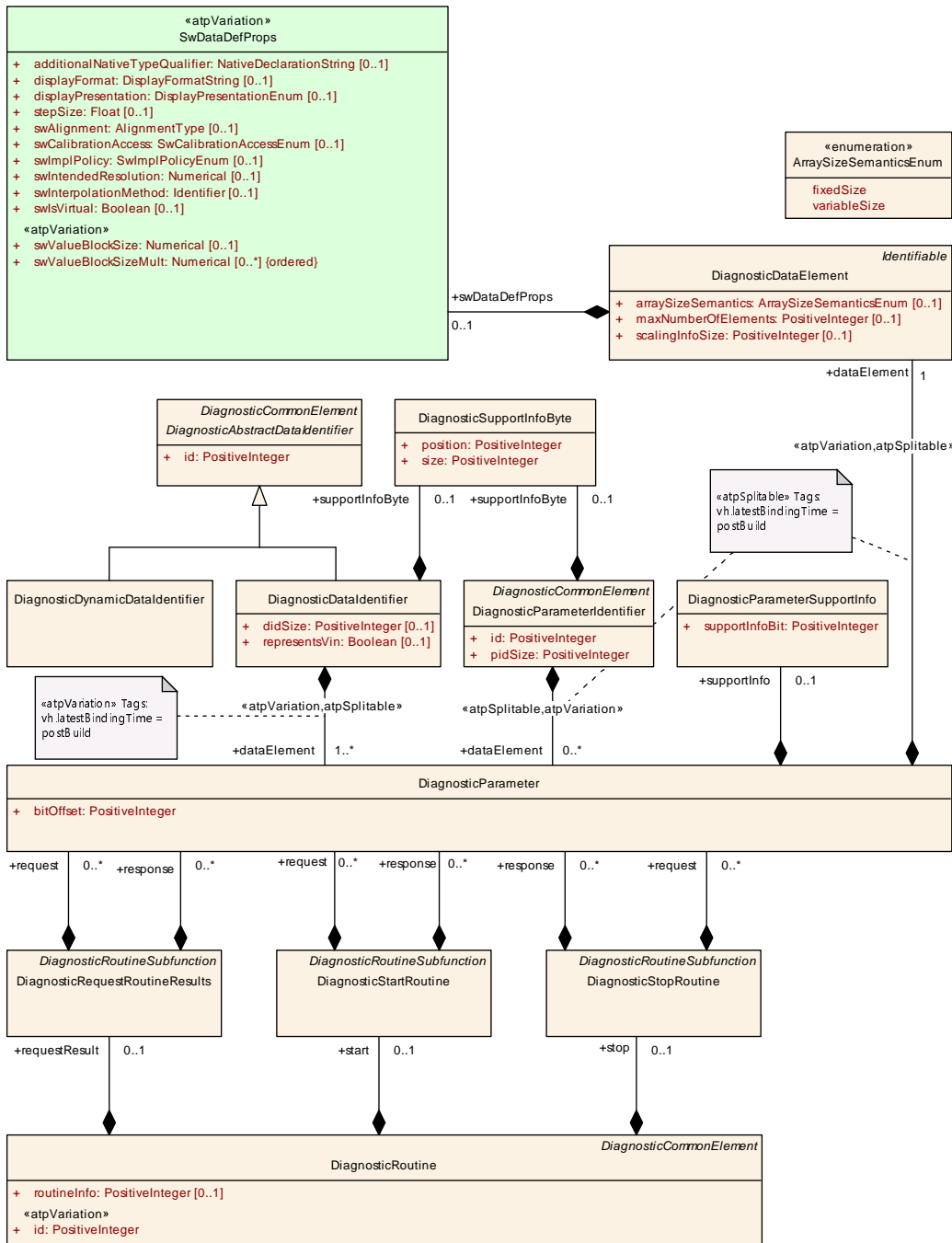


Figure 4.1: Common Diagnostic elements

<b>Class</b>	<i>DiagnosticCommonElement</i> (abstract)
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics
<b>Note</b>	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.
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>





<b>Class</b>	<i>DiagnosticCommonElement</i> (abstract)			
<b>Subclasses</b>	<a href="#">DiagnosticAbstractAliasEvent</a> , <a href="#">DiagnosticAbstractDataIdentifier</a> , <a href="#">DiagnosticAccessPermission</a> , <a href="#">DiagnosticAging</a> , <a href="#">DiagnosticCondition</a> , <a href="#">DiagnosticConditionGroup</a> , <a href="#">DiagnosticCustomServiceClass</a> , <a href="#">DiagnosticDataIdentifierSet</a> , <a href="#">DiagnosticEcuInstanceProps</a> , <a href="#">DiagnosticEnvironmentalCondition</a> , <a href="#">DiagnosticEvent</a> , <a href="#">DiagnosticExtendedDataRecord</a> , <a href="#">DiagnosticFimEventGroup</a> , <a href="#">DiagnosticFreezeFrame</a> , <a href="#">DiagnosticFunctionIdentifier</a> , <a href="#">DiagnosticFunctionIdentifierInhibit</a> , <a href="#">DiagnosticIndicator</a> , <a href="#">DiagnosticInfoType</a> , <a href="#">DiagnosticLumpr</a> , <a href="#">DiagnosticLumprDenominatorGroup</a> , <a href="#">DiagnosticLumprGroup</a> , <a href="#">DiagnosticJ1939</a> <a href="#">ExpandedFreezeFrame</a> , <a href="#">DiagnosticJ1939FreezeFrame</a> , <a href="#">DiagnosticJ1939Node</a> , <a href="#">DiagnosticJ1939Spn</a> , <a href="#">DiagnosticJ1939SwMapping</a> , <a href="#">DiagnosticMapping</a> , <a href="#">DiagnosticMeasurementIdentifier</a> , <a href="#">DiagnosticMemory</a> <a href="#">Destination</a> , <a href="#">DiagnosticMemoryIdentifier</a> , <a href="#">DiagnosticOperationCycle</a> , <a href="#">DiagnosticParameterIdentifier</a> , <a href="#">DiagnosticPowertrainFreezeFrame</a> , <a href="#">DiagnosticProtocol</a> , <a href="#">DiagnosticRoutine</a> , <a href="#">DiagnosticSecurityLevel</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">DiagnosticServiceTable</a> , <a href="#">DiagnosticSession</a> , <a href="#">DiagnosticTestResult</a> , <a href="#">DiagnosticTestRoutineIdentifier</a> , <a href="#">DiagnosticTroubleCode</a> , <a href="#">DiagnosticTroubleCode</a> <a href="#">Group</a> , <a href="#">DiagnosticTroubleCodeProps</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
-	-	-	-	-

**Table 4.1: DiagnosticCommonElement**

<b>Class</b>	<b>DiagnosticParameter</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
<b>Note</b>	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.			
<b>Base</b>	ARObject			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
bitOffset	PositiveInteger	1	attr	This represents the bitOffset of the DiagnosticParameter
dataElement	<a href="#">DiagnosticDataElement</a>	1	aggr	This represents the related dataElement of the Diagnostic Parameter  <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
supportInfo	<a href="#">DiagnosticParameterSupportInfo</a>	0..1	aggr	This attribute represents the ability to define which bit of the support info byte is representing this part of the PID.

**Table 4.2: DiagnosticParameter**

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 section 4.4.

The purpose of a `Data Identifier (DID)` is to associate a unique numerical value to 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 piece of data.

**[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.

|(RS\_DEXT\_00034, RS\_DEXT\_00035)

<b>Class</b>	<b>DiagnosticDataIdentifier</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
<b>Note</b>	This meta-class represents the ability to model a diagnostic data identifier (DID) that is fully specified regarding the payload at configuration-time. <b>Tags:</b> atp.recommendedPackage=DiagnosticDataIdentifiers			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticAbstractDataIdentifier</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataElement	<a href="#">DiagnosticParameter</a>	1..*	aggr	This is the dataElement associated with the Diagnostic DataIdentifier. <b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=dataElement, variationPoint.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	<a href="#">DiagnosticSupportInfoByte</a>	0..1	aggr	This attribute represents the supported information associated with the DiagnosticDataIdentifier.

**Table 4.3: DiagnosticDataIdentifier**

<b>Class</b>	<b>DiagnosticDynamicDataIdentifier</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
<b>Note</b>	This meta-class represents the ability to define a diagnostic data identifier (DID) at run-time. <b>Tags:</b> atp.recommendedPackage=DiagnosticDataIdentifiers			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticAbstractDataIdentifier</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
—	—	—	—	—

**Table 4.4: DiagnosticDynamicDataIdentifier**

<b>Class</b>	<b>DiagnosticAbstractDataIdentifier</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
<b>Note</b>	This meta-class represents an abstract base class for the modeling of a diagnostic data identifier (DID).			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">DiagnosticDataIdentifier</a> , <a href="#">DiagnosticDynamicDataIdentifier</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>





Class	DiagnosticAbstractDataIdentifier (abstract)			
id	PositiveInteger	1	attr	This is the numerical identifier used to identify the DiagnosticAbstractDataIdentifier in the scope of diagnostic workflow

**Table 4.5: DiagnosticAbstractDataIdentifier**

**[TPS\_DEXT\_01072] Purpose of 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. ]([RS\\_DEXT\\_00034](#))

**[constr\_1324] Existence of attribute [DiagnosticDataIdentifier.representsVin](#)** [ 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 build on top of an IP stack.

If the attribute exists and there is no IP used then the meaning of the attribute is simply irrelevant. Anyway, this situation should not be attributed to a misconfiguration.

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

The nature of such a `Data Element` could be compared to the nature of an `ISignal`<sup>1</sup> 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](#).

<b>Class</b>	<b>DiagnosticDataElement</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
<b>Note</b>	This meta-class represents the ability to describe a concrete piece of data to be taken into account for diagnostic purposes.			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>



<sup>1</sup>which represents the payload in “regular” bus communication



Class	DiagnosticDataElement			
arraySize Semantics	<a href="#">ArraySizeSemantics Enum</a>	0..1	attr	This attribute controls the meaning of the value of the array size.
maxNumberOfElements	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 DiagnosticDataElement if used with DiagnosticReadScalingDataBy Identifier
swDataDef Props	<a href="#">SwDataDefProps</a>	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.

**Table 4.6: DiagnosticDataElement**

**[constr\_1394] Value of [DiagnosticDataElement.maxNumberOfElements](#) depending on its existence** [ If the attribute [DiagnosticDataElement.maxNumberOfElements](#) exists then its value shall be greater than 0. ]()

**[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, or 32 and attribute [baseTypeEncoding](#) set to **either** NONE or 2C.

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]). ]([RS\\_DEXT\\_00034](#))

**[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 8 and attribute [baseTypeEncoding](#) set to NONE.

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. ]([RS\\_DEXT\\_00034](#))

**[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, or 32 and attribute [baseTypeEncoding](#) set to **either** NONE or 2C.

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. ] (*RS\_DEXT\_00034*)

**[constr\_1470] Value of `DiagnosticParameter.bitOffset`** [ The value of `DiagnosticParameter.bitOffset` shall only be set to a multiple of 8. ]()

Enumeration	ArraySizeSemanticsEnum
Package	M2::AUTOSARTemplates::CommonStructure::ImplementationDataTypes
Note	This type controls how the information about the number of elements in an <code>ApplicationArrayDataType</code> is to be interpreted.
Literal	<b>Description</b>
fixedSize	This means that the <code>ApplicationArrayDataType</code> will always have a fixed number of elements. <b>Tags:</b> atp.EnumerationValue=0
variableSize	This implies that the actual number of elements in the <code>ApplicationArrayDataType</code> might vary at run-time. The value of <code>arraySize</code> represents the maximum number of elements in the array. <b>Tags:</b> atp.EnumerationValue=1

**Table 4.7: ArraySizeSemanticsEnum**

The attribute `DiagnosticDataIdentifier.didSize` should be used if size of the DID data is intended to be enforced<sup>2</sup>.

**[constr\_1552] `DiagnosticDataIdentifier` referenced by `DiagnosticDataIdentifierSet`** [ If a `DiagnosticDataIdentifier` is referenced by `DiagnosticDataIdentifierSet` then the `DiagnosticDataIdentifier` shall not have gaps in between individual elements (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`). ]()

**[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. ] (*RS\_DEXT\_00034*)

**[constr\_1472] Existence of `DiagnosticDataIdentifier.supportInfoByte`** [ The attribute `DiagnosticDataIdentifier.supportInfoByte` shall not exist if the value of `DiagnosticDataIdentifier.id` is **outside the range 0xF400-0xF4FF**. ]()

<sup>2</sup>one use case for this enforcement was described by the removed [constr\_1471] contained in former versions of this document



### 4.2.1 Usage of SwDataDefProps

Please note that the definitions of properties like computation methods<sup>3</sup>, limits<sup>4</sup>, or units<sup>5</sup> of diagnostic data elements is based on shared resources of the AUTOSAR meta-model, namely by aggregation of meta-class `SwDataDefProps`.

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`.

**[constr\_1325] Allowed attributes of `SwDataDefProps` for `DiagnosticDataElement.swDataDefProps`** | The allowed attributes of `SwDataDefProps` for the aggregation in the role `DiagnosticDataElement.swDataDefProps` are defined in table 4.8. ]()

Attributes of <code>SwDataDefProps</code>	<code>DiagnosticDataElement.swDataDefProps</code>
<code>additionalNativeTypeQualifier</code>	N/A
<code>annotation</code>	N/A
<code>baseType.baseTypeDefinition.baseTypeEncoding</code>	D
<code>baseType.baseTypeDefinition.baseTypeSize</code>	D
<code>baseType.baseTypeDefinition.byteOrder</code>	D
<code>baseType.baseTypeDefinition.memAlignment</code>	N/A
<code>baseType.baseTypeDefinition.nativeDeclaration</code>	N/A
<code>compuMethod</code>	D
<code>dataConstr</code>	D
<code>displayFormat</code>	D
<code>displayPresentation</code>	N/A
<code>implementationDataType</code>	N/A
<code>invalidValue</code>	N/A
<code>swAddrMethod</code>	N/A
<code>swAlignment</code>	N/A
<code>swBitRepresentation</code>	N/A
<code>swCalibrationAccess</code>	N/A
<code>swCalprmAxisSet</code>	N/A
<code>swComparisonVariable</code>	N/A
<code>swDataDependency</code>	N/A
<code>swImplPolicy</code>	N/A
<code>swIntendedResolution</code>	N/A
<code>swInterpolationMethod</code>	N/A
<code>swIsVirtual</code>	N/A



<sup>3</sup>formalized as `CompuMethod` in AUTOSAR

<sup>4</sup>formalized as `DataConstr` in AUTOSAR

<sup>5</sup>formalized as `Unit` in AUTOSAR





Attributes of <code>SwDataDefProps</code>	<code>DiagnosticDataElement.swDataDefProps</code>
<code>swPointerTargetProps</code>	N/A
<code>swRecordLayout</code>	N/A
<code>swRefreshTiming</code>	N/A
<code>swTextProps</code>	N/A
<code>swValueBlockSize</code>	N/A
<code>unit</code>	D
<code>valueAxisDataType</code>	N/A

**Table 4.8: Allowed attributes of `SwDataDefProps` for `DiagnosticDataElement.swDataDefProps`**

The following legend applies to table 4.8:

Abbr.	Description
<b>D</b>	<b>Define</b> the attribute independent from settings to the left.
<b>I</b>	<b>Inherit</b> the definition from the left for usage in the scope of this element.
<b>N/A</b>	Attribute is <b>not applicable</b> for usage in the scope of this element.
<b>M</b>	Attribute is <b>meaningless</b> in the scope of this element. As it was allowed in previous versions, declaring it as Not Applicable (NA) would break compatibility. Tools shall ignore such an attribute without a warning.

**Table 4.9: Legend of table “Allowed attributes of `SwDataDefProps` for `DiagnosticDataElement.swDataDefProps`”**

Please note that, in comparison to similar tables appearing in other AUTOSAR documents (e.g. [8]), table 4.8 intentionally goes into more detail regarding the applicability of the attributes of `SwBaseType`. This is in contrast to similar tables contained in, e.g. the specification of the Software-Component Template [8]

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

<b>Class</b>	«atpVariation» <code>SwDataDefProps</code>
<b>Package</b>	M2::MSR::DataDictionary::DataDefProperties
<b>Note</b>	<p>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 <code>SwDataDefProps</code> is aggregated.</p> <p>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.</p> <p><code>SwDataDefProps</code> covers various aspects:</p> <ul style="list-style-type: none"> <li>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 Data Types in the programming language (or in AUTOSAR). This is mainly expressed by properties like <code>swRecordLayout</code> and <code>swCalprmAxisSet</code></li> </ul>





<b>Class</b>	«atpVariation» <b>SwDataDefProps</b>			
	<p style="text-align: center;">△</p> <ul style="list-style-type: none"> <li>Implementation aspects, mainly expressed by swImplPolicy, swVariableAccessImplPolicy, swAddrMethod, swPointerTargetProps, baseType, implementationDataType and additionalNativeTypeQualifier</li> <li>Access policy for the MCD system, mainly expressed by swCalibrationAccess</li> <li>Semantics of the data element, mainly expressed by compuMethod and/or unit, dataConstr, invalidValue</li> <li>Code generation policy provided by swRecordLayout</li> </ul> <p><b>Tags:</b> vh.latestBindingTime=codeGenerationTime</p>			
<b>Base</b>	ARObject			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
additionalNativeTypeQualifier	NativeDeclarationString	0..1	attr	<p>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.</p> <p><b>Tags:</b> xml.sequenceOffset=235</p>
annotation	Annotation	*	aggr	<p>This aggregation allows to add annotations (yellow pads ...) related to the current data object.</p> <p><b>Tags:</b> xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false</p>
baseType	<a href="#">SwBaseType</a>	0..1	ref	<p>Base type associated with the containing data object.</p> <p><b>Tags:</b> xml.sequenceOffset=50</p>
compuMethod	<a href="#">CompuMethod</a>	0..1	ref	<p>Computation method associated with the semantics of this data object.</p> <p><b>Tags:</b> xml.sequenceOffset=180</p>
dataConstr	<a href="#">DataConstr</a>	0..1	ref	<p>Data constraint for this data object.</p> <p><b>Tags:</b> xml.sequenceOffset=190</p>
displayFormat	DisplayFormatString	0..1	attr	<p>This property describes how a number is to be rendered e.g. in documents or in a measurement and calibration system.</p> <p><b>Tags:</b> xml.sequenceOffset=210</p>
displayPresentation	DisplayPresentationEnum	0..1	attr	<p>This attribute controls the presentation of the related data for measurement and calibration tools.</p>
implementationDataType	AbstractImplementationDataType	0..1	ref	<p>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</p> <ul style="list-style-type: none"> <li>redefinition of an ImplementationDataType via a "typedef" to another ImplementationDatatype</li> <li>the target type of a pointer (see SwPointerTargetProps), if it does not refer to a base type directly</li> </ul> <p style="text-align: center;">▽</p>





Class	«atpVariation» SwDataDefProps			
				<ul style="list-style-type: none"> <li>the data type of an array or record element within an ImplementationDataType, if it does not refer to a base type directly</li> <li>the data type of an SwServiceArg, if it does not refer to a base type directly</li> </ul> <p><b>Tags:</b> xml.sequenceOffset=215</p>
invalidValue	ValueSpecification	0..1	aggr	Optional value to express invalidity of the actual data element. <b>Tags:</b> 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. <b>Tags:</b> xml.sequenceOffset=30
swAlignment	AlignmentType	0..1	attr	The attribute describes the intended alignment of the DataPrototype. If the attribute is not defined the alignment is determined by the swBaseType size and the memory AllocationKeywordPolicy of the referenced SwAddr Method. <b>Tags:</b> xml.sequenceOffset=33
swBit Representation	SwBitRepresentation	0..1	aggr	Description of the binary representation in case of a bit variable. <b>Tags:</b> xml.sequenceOffset=60
swCalibration Access	SwCalibrationAccess Enum	0..1	attr	Specifies the read or write access by MCD tools for this data object. <b>Tags:</b> 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. <b>Tags:</b> xml.sequenceOffset=90
swComparison Variable	SwVariableRefProxy	*	aggr	Variables used for comparison in an MCD process. <b>Tags:</b> 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). <b>Tags:</b> 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. <b>Tags:</b> xml.sequenceOffset=220 xml.typeElement=false
swImplPolicy	SwImplPolicyEnum	0..1	attr	Implementation policy for this data object. <b>Tags:</b> 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



Class	«atpVariation» SwDataDefProps			
swInterpolationMethod	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. <b>Tags:</b> 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 . <b>Tags:</b> xml.sequenceOffset=260
swPointerTargetProps	SwPointerTargetProps	0..1	aggr	Specifies that the containing data object is a pointer to another data object. <b>Tags:</b> xml.sequenceOffset=280
swRecordLayout	SwRecordLayout	0..1	ref	Record layout for this data object. <b>Tags:</b> xml.sequenceOffset=290
swRefreshTiming	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. <b>Tags:</b> xml.sequenceOffset=300
swTextProps	SwTextProps	0..1	aggr	the specific properties if the data object is a text object. <b>Tags:</b> xml.sequenceOffset=120
swValueBlockSize	Numerical	0..1	attr	This represents the size of a Value Block <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime xml.sequenceOffset=80
swValueBlockSize Mult (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 swValueBlockSize shall be used and this attribute shall not exist. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime
unit	<a href="#">Unit</a>	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. <b>Tags:</b> xml.sequenceOffset=350





Class	«atpVariation» SwDataDefProps			
valueAxisDataType	<a href="#">ApplicationPrimitiveDataType</a>	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. <b>Tags:</b> xml.sequenceOffset=355

**Table 4.10: SwDataDefProps**

## 4.2.2 Definition of Arrays

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](#).

] ([RS\\_DEXT\\_00034](#), [RS\\_DEXT\\_00038](#))

**[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. ] ([RS\\_DEXT\\_00034](#), [RS\\_DEXT\\_00038](#))

**[constr\_1326] Existence of a variable-sized array** [ The value of the attribute [DiagnosticDataElement.arraySizeSemantics](#) **shall not** be set to [ArraySizeSemanticsEnum.variableSize](#) if the respective [DiagnosticDataElement](#) is referenced from a [DiagnosticServiceDataMapping](#). ]()

### 4.2.3 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 of the string on the ECU can be checked (after the establishment of a `DiagnosticMapping`) for a matching encoding specification.

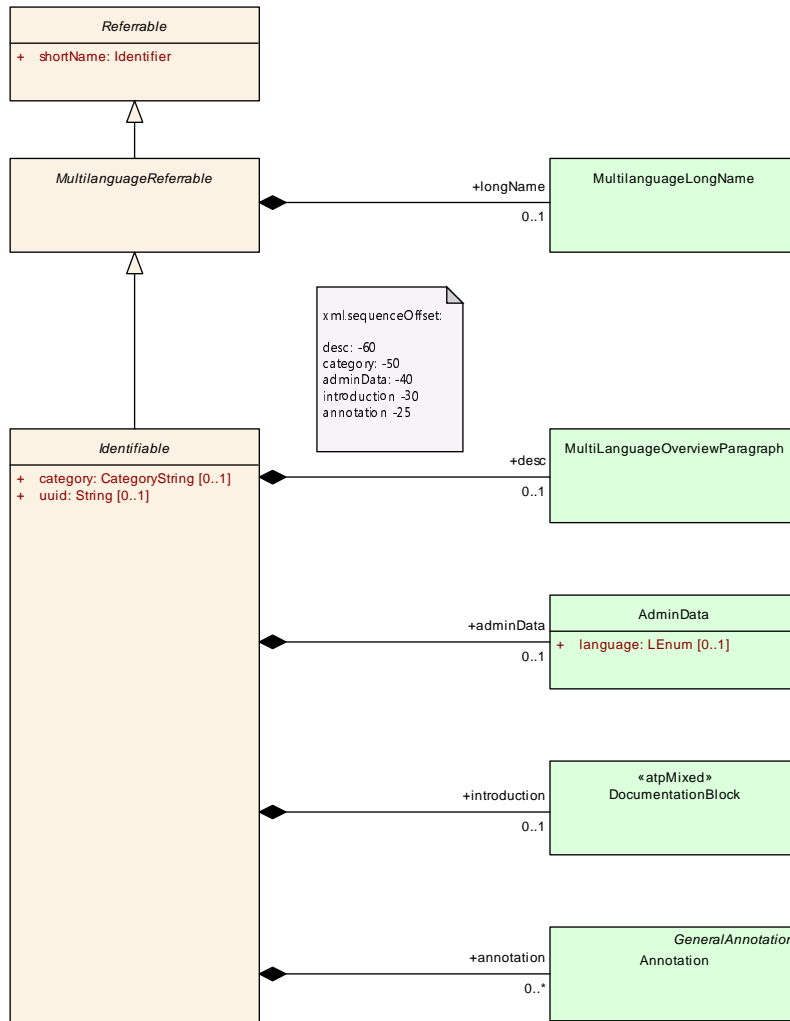
The encoding in the scope of the `DiagnosticDataElement` can be defined using the attribute `DiagnosticDataElement.swDataDefProps.baseType.baseType-Definition.baseTypeEncoding`.

## 4.3 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 4.2).

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

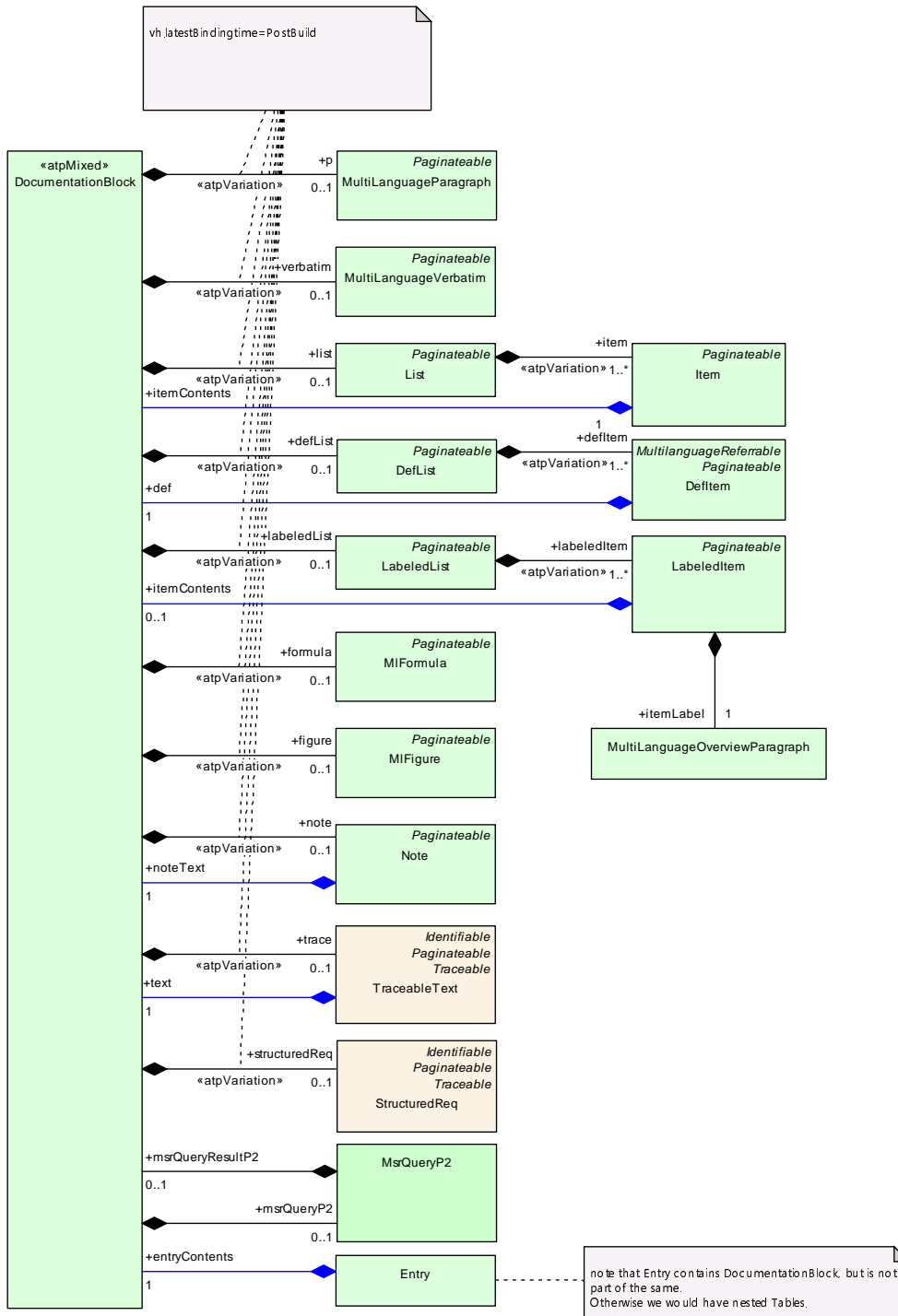
In other words, the technology described in Figures 4.2 and 4.3 is not limited to `DiagnosticDataElement` but has a much wider applicability in the context of the `DiagnosticExtract`.



**Figure 4.2: Definition of a textual description by means of `desc` and `introduction`**

The details regarding the specification of textual content that goes along a given diagnostics element is detailed in Figure 4.3. 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 4.2) to add short annotations (comparable to the usage of sticky notes) to diagnostic elements.



**Figure 4.3: Details of the modeling of DocumentationBlock**



<b>Class</b>	«atpMixed» <b>DocumentationBlock</b>			
<b>Package</b>	M2::MSR::Documentation::BlockElements			
<b>Note</b>	This class represents a documentation block. It is made of basic text structure elements which can be displayed in a table cell.			
<b>Base</b>	ARObject			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
defList	DefList	0..1	aggr	This represents a definition list in the documentation block. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=postBuild xml.sequenceOffset=40
figure	MIFigure	0..1	aggr	This represents a figure in the documentation block. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=postBuild xml.sequenceOffset=70
formula	MIFormula	0..1	aggr	This is a formula in the definition block. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=postBuild xml.sequenceOffset=60
labeledList	LabeledList	0..1	aggr	This represents a labeled list. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=postBuild xml.sequenceOffset=50
list	List	0..1	aggr	This represents numbered or unnumbered list. <b>Stereotypes:</b> atpVariation <b>Tags:</b> 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.
note	Note	0..1	aggr	This represents a note in the text flow. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=postBuild xml.sequenceOffset=80
p	MultiLanguage Paragraph	0..1	aggr	This is one particular paragraph. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=postBuild xml.sequenceOffset=10
structuredReq	<a href="#">StructuredReq</a>	0..1	aggr	This aggregation supports structured requirements embedded in a documentation block. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=postBuild xml.sequenceOffset=100
trace	<a href="#">TraceableText</a>	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. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=postBuild xml.sequenceOffset=90





<b>Class</b>	«atpMixed» DocumentationBlock			
verbatim	MultiLanguageVerbatim	0..1	aggr	This represents one particular verbatim text. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=postBuild xml.sequenceOffset=20

**Table 4.11: DocumentationBlock**

<b>Class</b>	MultiLanguageOverviewParagraph			
<b>Package</b>	M2::MSR::Documentation::TextModel::MultilanguageData			
<b>Note</b>	This is the content of a multilingual paragraph in an overview item.			
<b>Base</b>	ARObject			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
l2	LOverviewParagraph	1..*	aggr	This represents the text in one particular language. <b>Tags:</b> xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

**Table 4.12: MultiLanguageOverviewParagraph**

## 4.4 Diagnostic Contribution

[TPS\_DEXT\_01003] **DiagnosticContributionSet** is the central part of the **DiagnosticExtract** [ The central part of formalization of the concept of the **DiagnosticExtract** is the **DiagnosticContributionSet**.

To some extent, it fulfills a similar role as the **System** [6] in the communication domain. |(RS\_DEXT\_00001, RS\_DEXT\_00002)

The **DiagnosticContributionSet** maintains 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. `](RS_DEXT_00001, RS_DEXT_00002)`

**[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** `EcuInstances`. A `DiagnosticContributionSet` of `category` `DIAGNOSTICS_ECU_EXTRACT` shall be taken as the basis for the derivation of service software-components of the diagnostic stack.

`](RS_DEXT_00001, RS_DEXT_00002)`

**[constr\_1327] Multiplicity of `DiagnosticEcuInstanceProps.ecuInstance`** `[` 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`** `[` 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 refer to `EcuInstance` with the idea that both `DiagnosticContributionSet` and `DiagnosticServiceTable` can be modeled independently from 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.

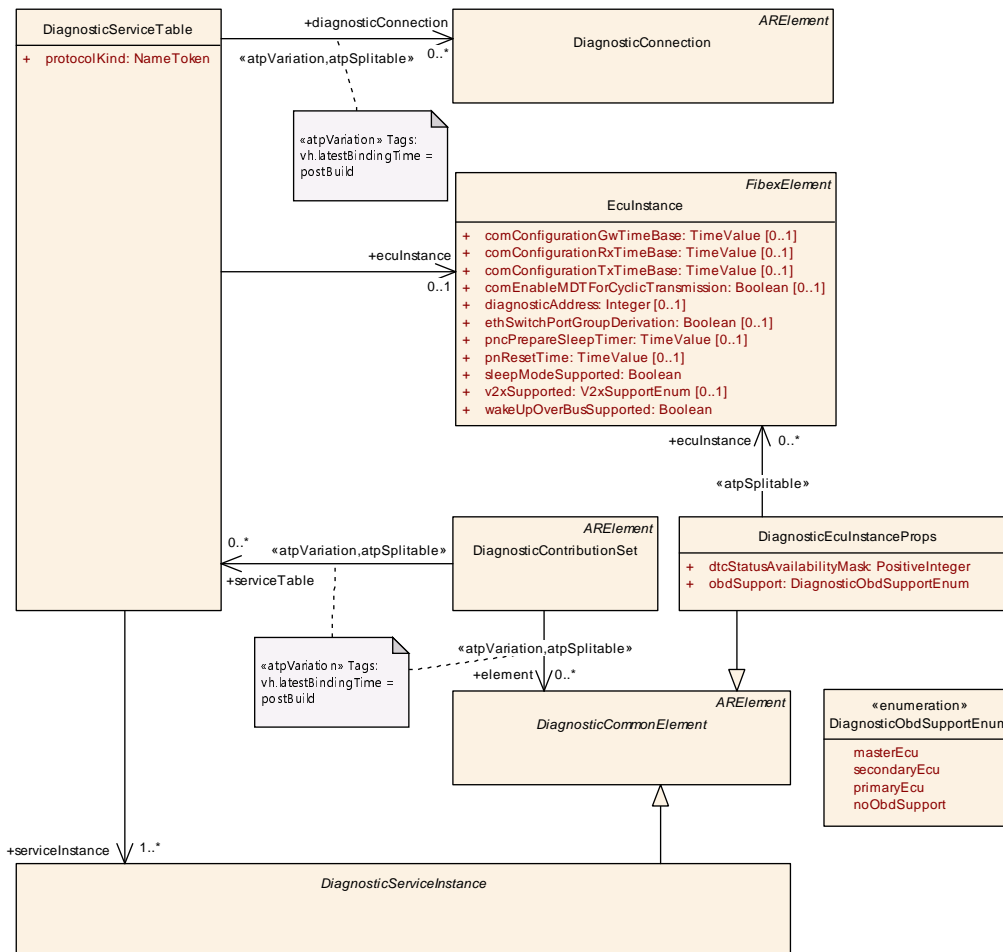


Figure 4.4: Diagnostic Contribution

[TPS\_DEXT\_01005] `DiagnosticContributionSet` can exist independently [ The `DiagnosticContributionSet` has been modeled as an `ARElement` so that its instances can exist independently from the existence of context-providing model-elements inside a given `ARPackage`. ] (*RS\_DEXT\_00001*, *RS\_DEXT\_00002*)

[TPS\_DEXT\_01005] elaborates on an important aspect that makes the `DiagnosticExtract` independent from 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.

<b>Class</b>	<b>DiagnosticContributionSet</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticContribution			
<b>Note</b>	<p>This meta-class represents a root node of a diagnostic extract. It bundles a given set of diagnostic model elements. The granularity of the DiagnosticContributionSet is arbitrary in order to support the aspect of decentralized configuration, i.e. different contributors can come up with an own DiagnosticContribution Set.</p> <p><b>Tags:</b> atp.recommendedPackage=DiagnosticContributionSets</p>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">AObject</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
common Properties	<a href="#">DiagnosticCommonProps</a>	0..1	aggr	<p>This attribute represents a collection of diagnostic properties that are shared among the entire DiagnosticContributionSet.</p> <p><b>Stereotypes:</b> atpSplittable <b>Tags:</b> atp.Splitkey=commonProperties</p>
element	<a href="#">DiagnosticCommonElement</a>	*	ref	<p>This represents a DiagnosticCommonElement considered in the context of the DiagnosticContributionSet</p> <p><b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=element, variationPoint.shortLabel vh.latestBindingTime=postBuild</p>
serviceTable	<a href="#">DiagnosticServiceTable</a>	*	ref	<p>This represents the collection of DiagnosticServiceTables to be considered in the scope of this DiagnosticContributionSet.</p> <p><b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=serviceTable, variationPoint.shortLabel vh.latestBindingTime=postBuild</p>

**Table 4.13: DiagnosticContributionSet**

## 4.5 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 as well as their priority. ] ([RS\\_DEXT\\_00059](#))

<b>Class</b>	<b>DiagnosticProtocol</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticContribution			
<b>Note</b>	<p>This meta-class represents the ability to define a diagnostic protocol.</p> <p><b>Tags:</b> atp.recommendedPackage=DiagnosticProtocols</p>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">AObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
diagnostic Connection	<a href="#">DiagnosticConnection</a>	*	ref	<p>This represents the collection of applicable DiagnosticConnections for this DiagnosticProtocol.</p> <p><b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=diagnosticConnection, variationPoint.shortLabel vh.latestBindingTime=postBuild</p>

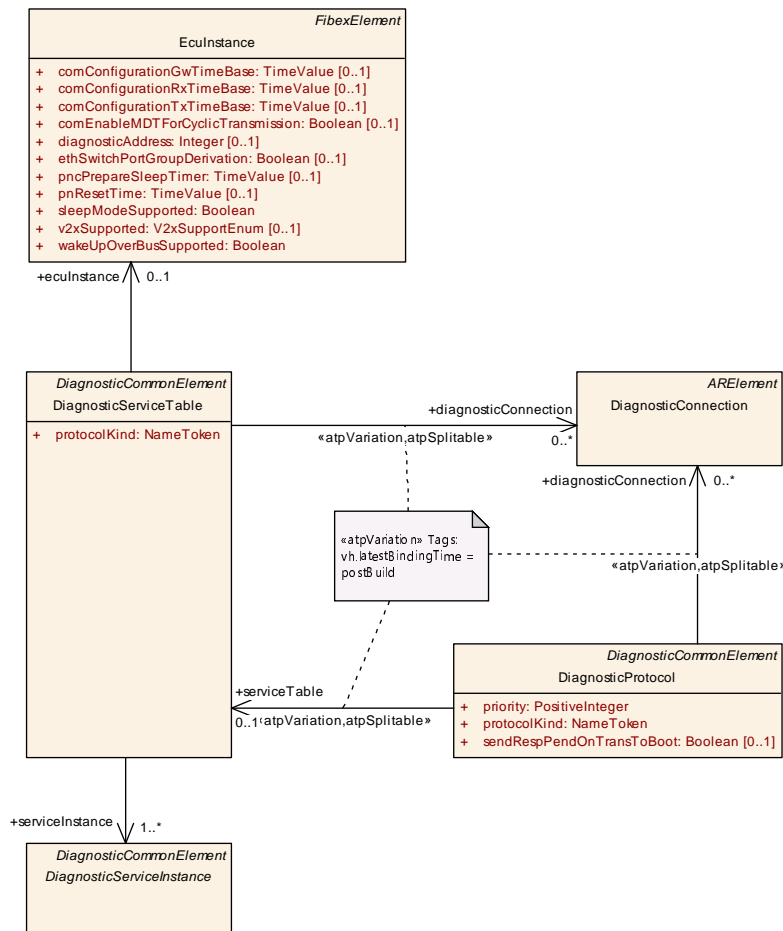




Class	DiagnosticProtocol			
priority	PositiveInteger	1	attr	This represents the priority of the diagnostic protocol in comparison to other diagnostic protocols. Lower numeric values represent higher protocol priority: <ul style="list-style-type: none"> <li>• 0 - Highest protocol priority</li> <li>• 255 - Lowest protocol priority</li> </ul>
protocolKind	NameToken	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").
serviceTable	DiagnosticServiceTable	0..1	ref	This represents the service table applicable for the given diagnostic protocol.  <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=serviceTable, variationPoint.shortLabel vh.latestBindingTime=postBuild

**Table 4.14: DiagnosticProtocol**

Each [DiagnosticProtocol](#) refers to at most one [DiagnosticServiceTable](#) and to a collection of [DiagnosticConnections](#).



**Figure 4.5: Modeling of DiagnosticProtocol**

Please note that both [DiagnosticServiceTable](#) and [DiagnosticProtocol](#) have an attribute named `protocolKind`.

<b>Class</b>	<b>DiagnosticServiceTable</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticContribution			
<b>Note</b>	This meta-class represents a model of a diagnostic service table, i.e. the UDS services applicable for a given ECU. <b>Tags:</b> atp.recommendedPackage=DiagnosticServiceTables			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
diagnosticConnection	<a href="#">DiagnosticConnection</a>	*	ref	This represents the <a href="#">DiagnosticConnection</a> that is taken for handling the data transmission for the enclosing <a href="#">DiagnosticServiceTable</a> .  It is possible to refer to more than one <a href="#">DiagnosticConnections</a> in order to support more than one diagnostic tester.  <b>Stereotypes:</b> atp.Splittable; atp.Variation <b>Tags:</b> atp.Splitkey=diagnosticConnection, variationPoint.shortLabel vh.latestBindingTime=postBuild
eculInstance	<a href="#">EculInstance</a>	0..1	ref	This represents the applicable <a href="#">EculInstance</a> for this <a href="#">DiagnosticServiceTable</a> .
protocolKind	<a href="#">NameToken</a>	1	attr	This identifies the applicable protocol.
serviceInstance	<a href="#">DiagnosticServiceInstance</a>	1..*	ref	This represents the collection of <a href="#">DiagnosticServiceInstances</a> to be considered in the scope of this <a href="#">DiagnosticServiceTable</a> ,

**Table 4.15: DiagnosticServiceTable**

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](#)** [ 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. ]  
([RS\\_DEXT\\_00039](#))

<b>Class</b>	<b>DiagnosticConnection</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::DiagnosticConnection			
<b>Note</b>	DiagnosticConncection that is used to describe the relationship between several TP connections. <b>Tags:</b> atp.recommendedPackage=DiagnosticConnections			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
functional Request	TpConnectionIdent	*	ref	Reference to functional request messages.
periodic ResponseUudt	PduTriggering	*	ref	Reference to UUDT responses.
physical Request	TpConnectionIdent	0..1	ref	Reference to a physical request message.
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.
responseOn Event	TpConnectionIdent	0..1	ref	Reference to a ROE message.

**Table 4.16: DiagnosticConnection**

Here is an example of a service table for UDS diagnostics:

**\$14** - GroupOfDTC: 0xFFFFFFFF

**\$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 [DiagnosticConnections](#)** [ In principle, the relation between a [DiagnosticServiceTable](#) and one or more [DiagnosticConnections](#) can be established in two possible ways:

- By means of the reference [DiagnosticServiceTable.diagnosticConnection](#). This way, the concept of protocols and their priorities relative to each other is **not** considered.
- By means of the references [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.

**[constr\_1406] `DiagnosticServiceTable.diagnosticConnection` vs. `DiagnosticProtocol.diagnosticConnection`** [ 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 `DiagnosticConnections` 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`. ]

()

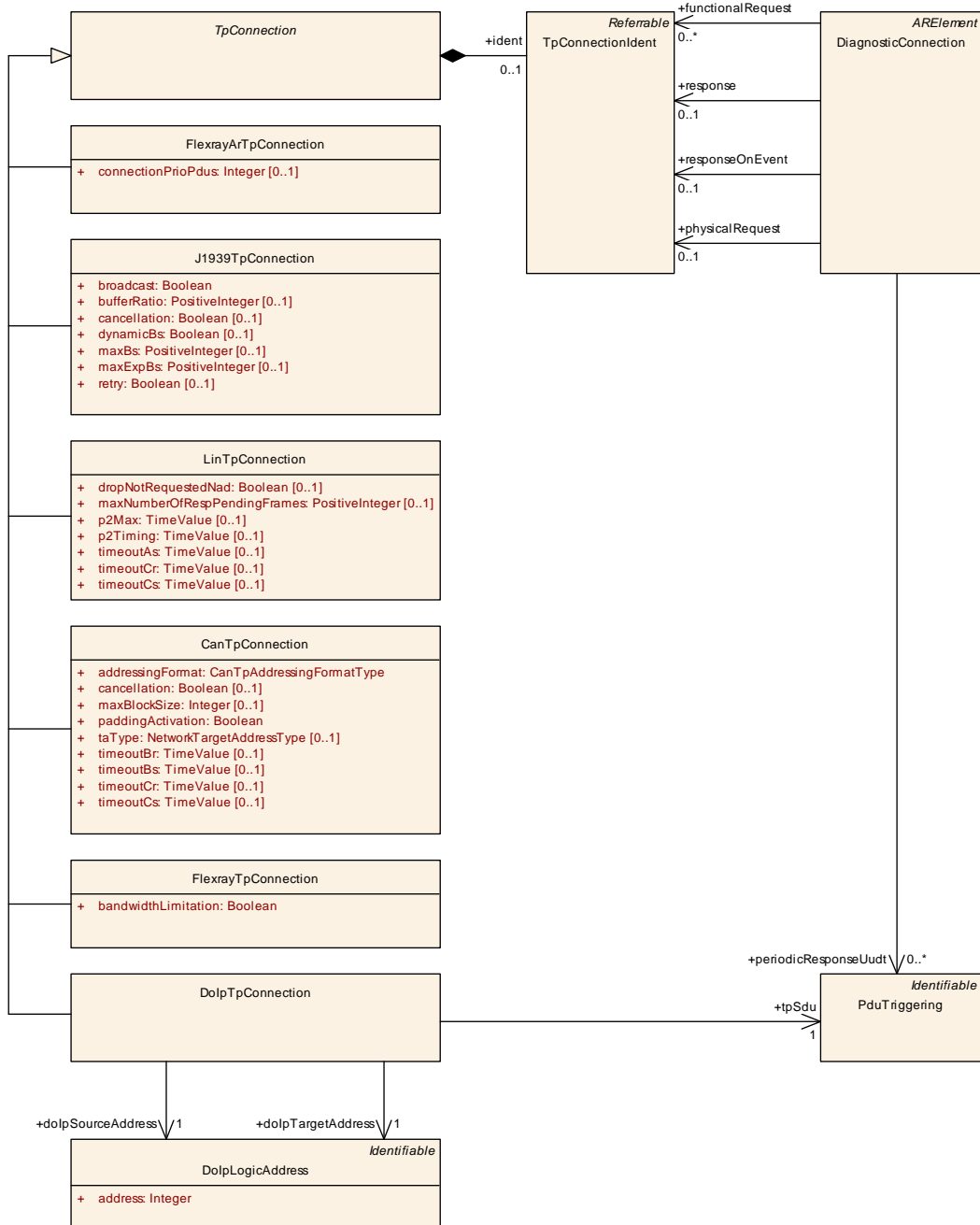


Figure 4.6: Modeling of DiagnosticConnection

## 4.6 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**. ](RS\_DEXT\_00001)

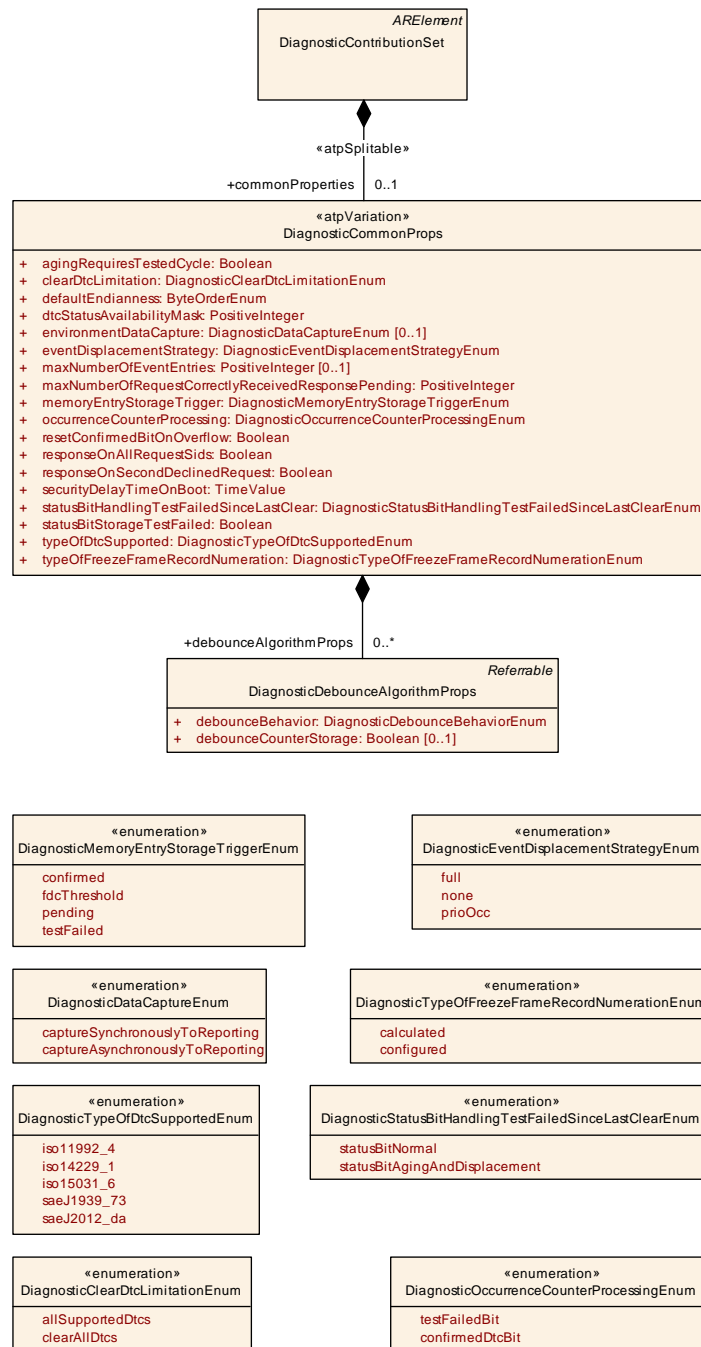


Figure 4.7: Common Diagnostic Properties

[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. ]([RS\\_DEXT\\_00001](#))

<b>Class</b>	«atpVariation» <b>DiagnosticCommonProps</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps			
<b>Note</b>	This meta-class aggregates a number of common properties that are shared among a diagnostic extract. <b>Tags:</b> vh.latestBindingTime=codeGenerationTime			
<b>Base</b>	ARObject			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
agingRequiresTestedCycle	Boolean	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.
clearDtcLimitation	<a href="#">DiagnosticClearDtcLimitationEnum</a>	1	attr	Defines the scope of the DEM_ClearDTC Api.
debounceAlgorithmProps	<a href="#">DiagnosticDebounceAlgorithmProps</a>	*	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..*.
defaultEndianness	ByteOrderEnum	1	attr	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.
dtcStatusAvailabilityMask	PositiveInteger	1	attr	Mask for the supported DTC status bits by the Dem.
environmentDataCapture	<a href="#">DiagnosticDataCaptureEnum</a>	0..1	attr	This attribute determines whether the capturing of environment data is done synchronously inside the report API function or whether the capturing shall be done asynchronously, i.e. after the report API function already terminated.
eventDisplacementStrategy	<a href="#">DiagnosticEventDisplacementStrategyEnum</a>	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.
maxNumberOfRequestCorrectlyReceivedResponsePending	PositiveInteger	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.
memoryEntryStorageTrigger	<a href="#">DiagnosticMemoryEntryStorageTriggerEnum</a>	1	attr	Describes the primary trigger to allocate an event memory entry.
occurrenceCounterProcessing	<a href="#">DiagnosticOccurrenceCounterProcessingEnum</a>	1	attr	This attribute defines the consideration of the fault confirmation process for the occurrence counter.
resetConfirmedBitOnOverflow	Boolean	1	attr	This attribute defines, whether the confirmed bit is reset or not while an event memory entry will be displaced.
responseOnAllRequestSids	Boolean	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).





<b>Class</b>	«atpVariation» <b>DiagnosticCommonProps</b>			
responseOnSecondDeclinedRequest	Boolean	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.
securityDelayTimeOnBoot	TimeValue	1	attr	Start delay timer on power on in seconds. This delay indicates the time at ECU boot power-on time where the Dcm remains in the default session and does not accept a security access.
statusBitHandlingTestFailedSinceLastClear	DiagnosticStatusBitHandlingTestFailedSinceLastClearEnum	1	attr	This attribute defines, whether the aging and displacement mechanism shall be applied to the "Test FailedSinceLastClear" status bits.
statusBitStorageTestFailed	Boolean	1	attr	This parameter is used to activate/deactivate the permanent storage of the "TestFailed" status bits. true: storage activated false: storage deactivated
typeOfDtcSupported	DiagnosticTypeOfDtcSupportedEnum	1	attr	This attribute defines the format returned by Dem_Dcm GetTranslationType and does not relate to/influence the supported Dem functionality.
typeOfFreezeFrameRecordNumeration	DiagnosticTypeOfFreezeFrameRecordNumerationEnum	1	attr	This attribute defines the type of assigning freeze frame record numbers for event-specific freeze frame records.

**Table 4.17: DiagnosticCommonProps**

<b>Enumeration</b>	<b>DiagnosticMemoryEntryStorageTriggerEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps
<b>Note</b>	Trigger types to allocate an event memory entry.
<b>Literal</b>	<b>Description</b>
confirmed	Status information of UDS DTC status bit 3 <b>Tags:</b> atp.EnumerationValue=0
fdcThreshold	Threshold to allocate an event memory entry and to capture the Freeze Frame. <b>Tags:</b> atp.EnumerationValue=1
pending	Status information of UDS DTC status bit 2. <b>Tags:</b> atp.EnumerationValue=2
testFailed	Status information of UDS DTC status bit 0. <b>Tags:</b> atp.EnumerationValue=3

**Table 4.18: DiagnosticMemoryEntryStorageTriggerEnum**

<b>Enumeration</b>	<b>DiagnosticDataCaptureEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps
<b>Note</b>	Data triggering types
<b>Literal</b>	<b>Description</b>
capture AsynchronouslyTo Reporting	This represents the intention to capture the environment data asynchronously after the actual capture API function terminated. <b>Tags:</b> atp.EnumerationValue=0
capture SynchronouslyTo Reporting	This represents the intention to capture the environment data synchronously within the capture API function. <b>Tags:</b> atp.EnumerationValue=1

**Table 4.19: DiagnosticDataCaptureEnum**

<b>Enumeration</b>	<b>DiagnosticTypeOfFreezeFrameRecordNumerationEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps
<b>Note</b>	FreezeFrame record numeration type
<b>Literal</b>	<b>Description</b>
calculated	Freeze frame records will be numbered consecutive starting by 1 in their chronological order. <b>Tags:</b> atp.EnumerationValue=0
configured	Freeze frame records will be numbered based on the given configuration in their chronological order. <b>Tags:</b> atp.EnumerationValue=1

**Table 4.20: DiagnosticTypeOfFreezeFrameRecordNumerationEnum**

<b>Enumeration</b>	<b>DiagnosticClearDtcLimitationEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps
<b>Note</b>	Scope of the DEM_ClearDTC Api.
<b>Literal</b>	<b>Description</b>
allSupportedDtc	DEM_ClearDtc API accepts all supported DTC values. <b>Tags:</b> atp.EnumerationValue=0
clearAllDtc	DEM_ClearDtc API accepts ClearAllDTCs only. <b>Tags:</b> atp.EnumerationValue=1

**Table 4.21: DiagnosticClearDtcLimitationEnum**

<b>Enumeration</b>	<b>DiagnosticEventDisplacementStrategyEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps
<b>Note</b>	Defines the displacement strategy.
<b>Literal</b>	<b>Description</b>
full	Event memory entry displacement is enabled, by consideration of priority active/passive status, and occurrence. <b>Tags:</b> atp.EnumerationValue=0
none	Event memory entry displacement is disabled. <b>Tags:</b> atp.EnumerationValue=1
prioOcc	Event memory entry displacement is enabled, by consideration of priority and occurrence (but without active/passive status). <b>Tags:</b> atp.EnumerationValue=2

**Table 4.22: DiagnosticEventDisplacementStrategyEnum**

<b>Enumeration</b>	<b>DiagnosticOccurrenceCounterProcessingEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps
<b>Note</b>	The occurrence counter triggering types.
<b>Literal</b>	<b>Description</b>
confirmedDtcBit	The occurrence counter is triggered by the TestFailed bit if the fault confirmation was successful (ConfirmedDTC bit is set). <b>Tags:</b> atp.EnumerationValue=0
testFailedBit	The occurrence counter is only triggered by the TestFailed bit (and the fault confirmation is not considered). <b>Tags:</b> atp.EnumerationValue=1

**Table 4.23: DiagnosticOccurrenceCounterProcessingEnum**

<b>Enumeration</b>	<b>DiagnosticTypeOfDtcSupportedEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps
<b>Note</b>	Supported Dtc Types
<b>Literal</b>	<b>Description</b>
iso11992_4	ISO11992-4 DTC format <b>Tags:</b> atp.EnumerationValue=0
iso14229_1	ISO14229-1 DTC format (3 byte format) <b>Tags:</b> atp.EnumerationValue=1
iso15031_6	ISO15031-6 DTC format (2 byte format) <b>Tags:</b> atp.EnumerationValue=2
saeJ1939_73	SAEJ1939-73 DTC format <b>Tags:</b> atp.EnumerationValue=3
saeJ2012_da	SAE_J2012-DA_DTCFormat_00 (3 byte format) <b>Tags:</b> atp.EnumerationValue=4

**Table 4.24: DiagnosticTypeOfDtcSupportedEnum**

## 5 Diagnostic Services

### 5.1 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 ISO 14229 [16].

### 5.2 Service Instance vs. 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 `DiagnosticServiceInstance`) implements a concrete use of a diagnostic service in a given context whereas purpose of the diagnostic service *class* (formalized as `DiagnosticServiceClass`) is to provide properties that are shared among all existing diagnostic service *instances* in the model.

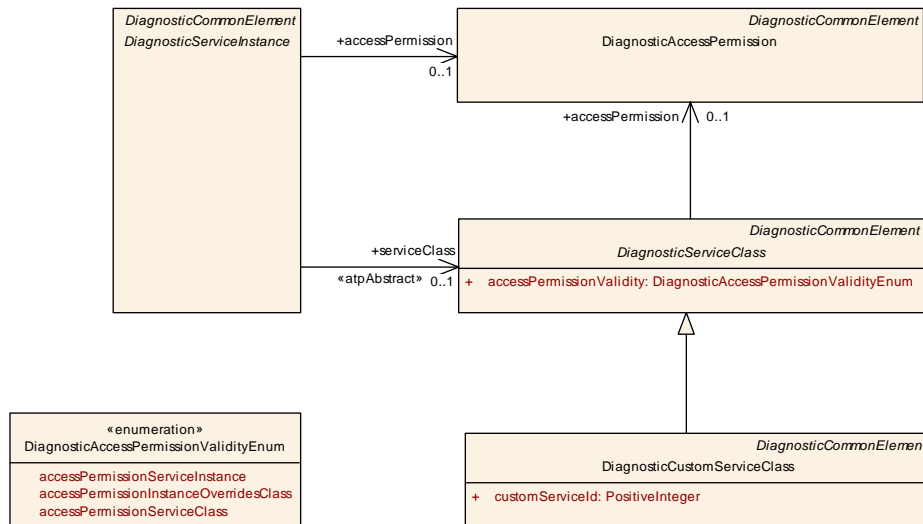


Figure 5.1: Common service elements

**[constr\_1329] Existence of concrete sub-classes of `DiagnosticServiceClass` in the context created by a `DiagnosticContributionSet`** [ 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` shall only appear once in the context created by a `DiagnosticContributionSet`
- If the subclass of `DiagnosticServiceClass` appears multiple times in the context created by a `DiagnosticContributionSet` then all instances shall have identical values for all of their attributes.

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

]()

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.



<b>Class</b>	<i>DiagnosticServiceClass</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommonService			
<b>Note</b>	This meta-class provides the ability to define common properties that are shared among all instances of sub-classes of DiagnosticServiceInstance.			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">DiagnosticClearDiagnosticInformationClass</a> , <a href="#">DiagnosticClearResetEmissionRelatedInfoClass</a> , <a href="#">DiagnosticComControlClass</a> , <a href="#">DiagnosticControlDTCSettingClass</a> , <a href="#">DiagnosticCustomServiceClass</a> , <a href="#">DiagnosticDataTransferClass</a> , <a href="#">DiagnosticDynamicallyDefineDataIdentifierClass</a> , <a href="#">DiagnosticEcuResetClass</a> , <a href="#">DiagnosticIOControlClass</a> , <a href="#">DiagnosticReadDTCInformationClass</a> , <a href="#">DiagnosticReadDataByIdentifierClass</a> , <a href="#">DiagnosticReadDataByPeriodicIDClass</a> , <a href="#">DiagnosticReadMemoryByAddressClass</a> , <a href="#">DiagnosticReadScalingDataByIdentifierClass</a> , <a href="#">DiagnosticRequestControlOfOnBoardDeviceClass</a> , <a href="#">DiagnosticRequestCurrentPowertrainDataClass</a> , <a href="#">DiagnosticRequestEmissionRelatedDTCClass</a> , <a href="#">DiagnosticRequestEmissionRelatedDTCPermanentStatusClass</a> , <a href="#">DiagnosticRequestFileTransferClass</a> , <a href="#">DiagnosticRequestOnBoardMonitoringTestResultsClass</a> , <a href="#">DiagnosticRequestPowertrainFreezeFrameDataClass</a> , <a href="#">DiagnosticRequestUploadClass</a> , <a href="#">DiagnosticRequestVehicleInfoClass</a> , <a href="#">DiagnosticResponseOnEventClass</a> , <a href="#">DiagnosticRoutineControlClass</a> , <a href="#">DiagnosticSecurityAccessClass</a> , <a href="#">DiagnosticSessionControlClass</a> , <a href="#">DiagnosticTransferExitClass</a> , <a href="#">DiagnosticWriteDataByIdentifierClass</a> , <a href="#">DiagnosticWriteMemoryByAddressClass</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
access Permission	<a href="#">DiagnosticAccessPermission</a>	0..1	ref	This represents the collection of DiagnosticAccessPermissions that allow for the execution of the referencing DiagnosticServiceClass.
access Permission Validity	<a href="#">DiagnosticAccessPermissionValidityEnum</a>	1	attr	This attribute is responsible for clarifying the validity of the accessPermission reference.

**Table 5.1: DiagnosticServiceClass**

<b>Class</b>	<i>DiagnosticServiceInstance</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommonService			
<b>Note</b>	This represents a concrete instance of a diagnostic service.			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">DiagnosticClearDiagnosticInformation</a> , <a href="#">DiagnosticClearResetEmissionRelatedInfo</a> , <a href="#">DiagnosticComControl</a> , <a href="#">DiagnosticControlDTCSetting</a> , <a href="#">DiagnosticDataByIdentifier</a> , <a href="#">DiagnosticDynamicallyDefineDataIdentifier</a> , <a href="#">DiagnosticEcuReset</a> , <a href="#">DiagnosticIOControl</a> , <a href="#">DiagnosticMemoryByAddress</a> , <a href="#">DiagnosticReadDTCInformation</a> , <a href="#">DiagnosticReadDataByPeriodicID</a> , <a href="#">DiagnosticRequestControlOfOnBoardDevice</a> , <a href="#">DiagnosticRequestCurrentPowertrainData</a> , <a href="#">DiagnosticRequestEmissionRelatedDTC</a> , <a href="#">DiagnosticRequestEmissionRelatedDTCPermanentStatus</a> , <a href="#">DiagnosticRequestFileTransfer</a> , <a href="#">DiagnosticRequestOnBoardMonitoringTestResults</a> , <a href="#">DiagnosticRequestPowertrainFreezeFrameData</a> , <a href="#">DiagnosticRequestVehicleInfo</a> , <a href="#">DiagnosticResponseOnEvent</a> , <a href="#">DiagnosticRoutineControl</a> , <a href="#">DiagnosticSecurityAccess</a> , <a href="#">DiagnosticSessionControl</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
access Permission	<a href="#">DiagnosticAccessPermission</a>	0..1	ref	This represents the collection of DiagnosticAccessPermissions that allow for the execution of the referencing DiagnosticServiceInstance..
serviceClass	<a href="#">DiagnosticServiceClass</a>	0..1	ref	<p>This represents the corresponding "class", i.e. this meta-class provides properties that are shared among all instances of applicable sub-classes of DiagnosticServiceInstance.</p> <p>The subclasses that affected by this pattern implement references to the applicable "class"-role that substantiate this abstract reference.</p> <p><b>Stereotypes:</b> atpAbstract</p>

**Table 5.2: DiagnosticServiceInstance**

**[TPS\_DEXT\_01009] Limited support for the configuration of custom diagnostic services** [ Beside the support for explicitly modeled diagnostic services, there is also a limited support for the configuration of custom diagnostic services.

The formalization, however, goes only so far as to define the `DiagnosticCustomServiceClass` and its attribute `customServiceId` that allows for the definition of the custom service identifier used for the purpose. ]([RS\\_DEXT\\_00047](#))

**[constr\_1330] Custom service identifier shall not overlap with standardized service identifiers** [ The value of the attribute `customServiceId` shall not be set to any of the values reserved for standardized service identifiers as defined by the ISO 14229-1, see [16]. ]()

**[TPS\_DEXT\_01010] Configuration of custom diagnostic services** [ The support for the configuration of custom diagnostic services within the `DiagnosticExtract` does not extend beyond the ability to define that attribute `DiagnosticCustomServiceClass.customServiceId`.

There is no corresponding formalization of a diagnostic service instance that supports references to e.g. a `DiagnosticDataIdentifier`. ]([RS\\_DEXT\\_00047](#))

<b>Class</b>	<b>DiagnosticCustomServiceClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommonService			
<b>Note</b>	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. <b>Tags:</b> atp.recommendedPackage=DiagnosticCustomServiceClasses			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
customServiceId	PositiveInteger	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 5.3: DiagnosticCustomServiceClass**

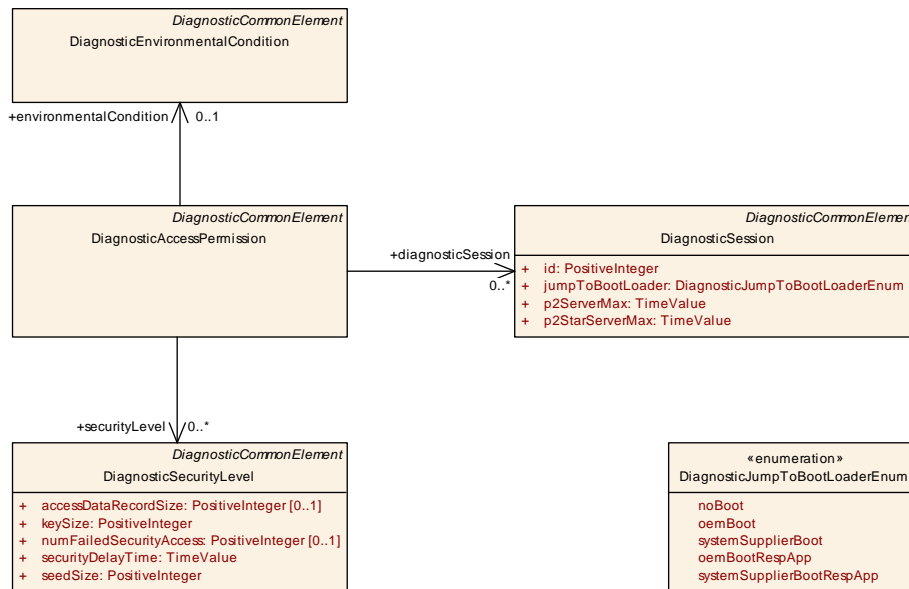
## 5.3 Access Permission, Session, Security Level

This chapter discusses a set of meta-classes that have been created to represent the concept of an *access permission* used in the context of the `Dcm`.

### 5.3.1 Introduction to Access Permission

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 5.2).

At runtime, [DiagnosticSessions](#) are used to create a context for the execution of diagnostic functionality. Servers usually support a variety of different [DiagnosticSessions](#). It is possible to switch between [DiagnosticSessions](#) at runtime.



**Figure 5.2: Common model elements relevant for 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`

in terms of how access permission is granted is subject to the specification of the Dcm [10]. ] ([RS\\_DEXT\\_00040](#))

**[TPS\_DEXT\_01011] Semantics of [DiagnosticSession.id](#)** [ The value of the attribute [DiagnosticSession.id](#) has a given semantics according to ISO 14229-1 [16]. 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*.

The value range **0x40 .. 0x5F** is reserved for manufacturer-specific use. ]  
([RS\\_DEXT\\_00040](#))

<b>Class</b>	<b>DiagnosticSecurityLevel</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm			
<b>Note</b>	This meta-class represents the ability to define a security level considered for diagnostic purposes. <b>Tags:</b> atp.recommendedPackage=DiagnosticSecurityLevels			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
accessDataRecordSize	PositiveInteger	0..1	attr	This represents the size of the AccessDataRecord used in GetSeed. Unit:byte.
keySize	PositiveInteger	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	1	attr	This represents the delay time after a failed security access. Unit: second.
seedSize	PositiveInteger	1	attr	This represents the size of the security seed. Unit: byte.

**Table 5.4: DiagnosticSecurityLevel**

[TPS\_DEXT\_01012] **Rationale for the modeling of the multiplicity of [DiagnosticAccessPermission.securityLevel](#)** [ The multiplicity of [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 [DiagnosticSecurityLevels](#) required. As a consequence, the execution of the [DiagnosticServiceInstance](#) that references the given [DiagnosticAccessPermission](#) is always possible.

] ([RS\\_DEXT\\_00041](#), [RS\\_DEXT\\_00042](#))

[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 4.3. An example of how the definition of pre- and run-conditions may look like in ARXML is sketched in listing 5.1.

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](#).

]([RS\\_DEXT\\_00041](#), [RS\\_DEXT\\_00045](#))

**Listing 5.1: Example for the definition of pre- and run-conditions for [DiagnosticAccessPermission](#)**

```
<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>
```

[constr\_1419] Value of [DiagnosticSecurityLevel.accessDataRecordSize](#) [ If the attribute [DiagnosticSecurityLevel.accessDataRecordSize](#) exists then its value shall be greater than zero. ]()

<b>Class</b>	<b>DiagnosticAccessPermission</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm			
<b>Note</b>	<p>This represents the specification of whether a given service can be accessed according to the existence of meta-classes referenced by a particular DiagnosticAccessPermission.</p> <p>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.</p> <p><b>Tags:</b> atp.recommendedPackage=DiagnosticAccessPermissions</p>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
diagnosticSession	<a href="#">DiagnosticSession</a>	*	ref	This represents the associated DiagnosticSessions
environmentalCondition	<a href="#">DiagnosticEnvironmentalCondition</a>	0..1	ref	This represents the environmental conditions associated with the access permission.
securityLevel	<a href="#">DiagnosticSecurityLevel</a>	*	ref	This represents the associated DiagnosticSecurityLevels

**Table 5.5: DiagnosticAccessPermission**

<b>Class</b>	<b>DiagnosticSession</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm			
<b>Note</b>	<p>This meta-class represents the ability to define a diagnostic session.</p> <p><b>Tags:</b> atp.recommendedPackage=DiagnosticSessions</p>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
id	PositiveInteger	1	attr	This is the numerical identifier used to identify the DiagnosticSession in the scope of diagnostic workflow
jumpToBootLoader	<a href="#">DiagnosticJumpToBootLoaderEnum</a>	1	attr	<p>This attribute represents the ability to define whether this diagnostic session allows to jump to Bootloader (OEM Bootloader or System Supplier Bootloader).</p> <p>If this diagnostic session doesn't allow to jump to Bootloader the value JumpToBootLoaderEnum.noBoot shall be chosen.</p>
p2ServerMax	TimeValue	1	attr	<p>This is the session value for P2ServerMax in seconds (per Session Control).</p> <p>The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds.</p>
p2StarServerMax	TimeValue	1	attr	<p>This is the session value for P2*ServerMax in seconds (per Session Control).</p> <p>The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds.</p>

**Table 5.6: DiagnosticSession**

<b>Enumeration</b>	<b>DiagnosticJumpToBootLoaderEnum</b>		
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm		
<b>Note</b>	This enumeration contains the options for jumping to a boot loader.		
<b>Literal</b>	<b>Description</b>		
noBoot	<p>This diagnostic session doesn't allow to jump to Bootloader.</p> <p><b>Tags:</b> atp.EnumerationValue=0</p>		





Enumeration	DiagnosticJumpToBootLoaderEnum
oemBoot	This diagnostic session allows to jump to OEM Bootloader. In this case the bootloader send the final response. <b>Tags:</b> atp.EnumerationValue=1
oemBootRespApp	This diagnostic session allows to jump to OEM Bootloader and application sends final response. <b>Tags:</b> atp.EnumerationValue=3
systemSupplierBoot	This diagnostic session allows to jump to System Supplier Bootloader. In this case the bootloader send the final response. <b>Tags:</b> atp.EnumerationValue=2
systemSupplierBootRespApp	This diagnostic session allows to jump to System Supplier Bootloader and application sends final response. <b>Tags:</b> atp.EnumerationValue=4

**Table 5.7: DiagnosticJumpToBootLoaderEnum**

### 5.3.2 Prioritization of Access Permission

The definition of *access permission* itself can be done on different levels. It is therefore necessary to define how the existence of *access permission* on these different levels is supposed to be interpreted.

#### [TPS\_DEXT\_01061] Supported scenarios for the definition of *access permission*

The following scenarios are possible for the definition of *access permission*:

- The *access permission* is defined on the level of a `DiagnosticServiceClass`. In this scenario, the **intended semantics** is that this configuration is binding for **all** `DiagnosticServiceInstances` derived from the `DiagnosticServiceClass`.

The configuration of a `DiagnosticServiceInstance.accessPermission` is considered as an error and shall be reported accordingly.

This scenario applies if `DiagnosticServiceClass.accessPermissionValidity` is set to the value `accessPermissionServiceClass`.

- The *access permission* is defined on the level of an individual `DiagnosticServiceInstance`. In this scenario, the **intended semantics** is that the `DiagnosticServiceClass` shall **not make any assumptions about the definition of the applicable *access permission***.

The configuration of a `DiagnosticServiceClass.accessPermission` is considered as an error and shall be reported accordingly. This scenario applies if `DiagnosticServiceClass.accessPermissionValidity` is set to the value `accessPermissionServiceInstance`.

- The definition of both `DiagnosticServiceClass.accessPermission` and `DiagnosticServiceInstance.accessPermission` is positively allowed.



In this scenario, the **intended semantics** is that if `DiagnosticServiceClass.accessPermission` exists the individual `DiagnosticServiceInstance.accessPermission` **are not required to define** `DiagnosticServiceInstance.accessPermission` **but if they do** then the `DiagnosticServiceInstance.accessPermission` **gets priority** over the definition of `DiagnosticServiceClass.accessPermission`.

This basically boils down to the ability to **override** the setting for *access permission* made on the level of a `DiagnosticServiceClass` by the setting on the level of an `DiagnosticServiceInstance`.

At the same time, this scenario saves some file footprint because (given the existence of `DiagnosticServiceClass.accessPermission`) there is no need to define individual `DiagnosticServiceInstance.accessPermission` unless there is a dedicated need for them.

This scenario applies if `DiagnosticServiceClass.accessPermissionValidity` is set to the value `accessPermissionInstanceOverridesClass`.

|(RS\_DEXT\_00041, RS\_DEXT\_00049, RS\_DEXT\_00050)

The scenarios defined by [TPS\_DEXT\_01061] need modeling support in order to allow the user to precisely express the intended semantics of a model with respect to *access permission*. For this purpose the attribute `DiagnosticServiceClass.accessPermissionValidity` is available.

Enumeration	DiagnosticAccessPermissionValidityEnum
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommonService
Note	This meta-class provides settings for how the accessPermission shall be resolved between DiagnosticServiceInstance and DiagnosticServiceClass.
Literal	<b>Description</b>
accessPermissionInstanceOverridesClass	This means that accessPermission set at the DiagnosticServiceInstance will override the accessPermission defined at the DiagnosticServiceClass. <b>Tags:</b> atp.EnumerationValue=1
accessPermissionServiceClass	This means that the DiagnosticServiceClass is in charge to define the accessPermission. <b>Tags:</b> atp.EnumerationValue=2
accessPermissionServiceInstance	This means that the DiagnosticServiceInstance is in charge of defining the accessPermission <b>Tags:</b> atp.EnumerationValue=3

**Table 5.8: DiagnosticAccessPermissionValidityEnum**

[TPS\_DEXT\_01062] **Existence of `DiagnosticServiceClass.accessPermissionValidity` in an incomplete model** | If the attribute `DiagnosticServiceClass.accessPermissionValidity` does not exist then it shall be assumed that the configuration is incomplete. |(RS\_DEXT\_00001, RS\_DEXT\_00002, RS\_DEXT\_00041)

Please note that the model state described in [TPS\_DEXT\_01062] is still allowed because it may only be possible to decide about the value of the attribute at later points in the model's life cycle.



[TPS\_DEXT\_01063] **Existence of `DiagnosticServiceClass.accessPermissionValidity` in a complete model** [ As the model's life cycle approaches the point where the model is considered complete the attribute `DiagnosticServiceClass.accessPermissionValidity` shall exist in order to be able to properly figure out the intended model semantics. ]([RS\\_DEXT\\_00001](#), [RS\\_DEXT\\_00002](#), [RS\\_DEXT\\_00041](#))

[TPS\_DEXT\_01052] **Existence of attribute `DiagnosticServiceInstance.accessPermission`** [ Regarding the existence of the attribute `DiagnosticServiceInstance.accessPermission` the following rules apply:

- If neither the attribute `DiagnosticServiceInstance.accessPermission` or `DiagnosticServiceClass.accessPermission` exist it is assumed that the configuration is incomplete as no access permission is defined.
- If either the attribute `DiagnosticServiceInstance.accessPermission` or `DiagnosticServiceClass.accessPermission` exists but does not have further references to `DiagnosticSession`, `DiagnosticEnvironmentalCondition`, or `DiagnosticSecurityLevel` then this means that the affected diagnostic services can be executed in any diagnostic session or security level. In other words, no restriction applies.

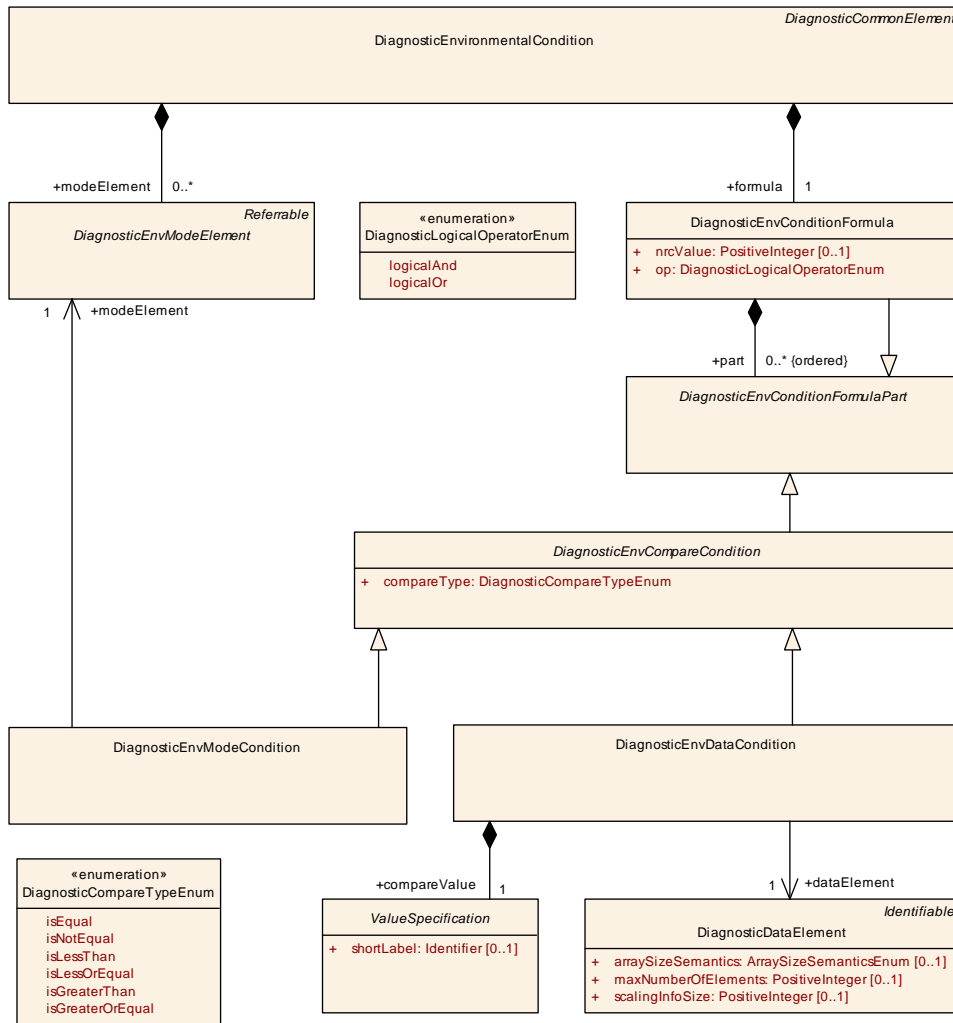
]([RS\\_DEXT\\_00041](#), [RS\\_DEXT\\_00049](#))

## 5.4 Environmental Conditions for the Execution of Diagnostic Services

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.



**Figure 5.3: Formal modeling of the consideration of environmental 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`.

<b>Class</b>	<b>DiagnosticEnvironmentalCondition</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
<b>Note</b>	The meta-class <code>DiagnosticEnvironmentalCondition</code> 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. <code>vehicle speed == 0</code> ). <b>Tags:</b> <code>atp.recommendedPackage=DiagnosticEnvironmentalConditions</code>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
formula	<a href="#">DiagnosticEnvConditionFormula</a>	1	aggr	This attribute represents the formula part of the <code>DiagnosticEnvironmentalCondition</code> .





Class	DiagnosticEnvironmentalCondition			
modeElement	<a href="#">DiagnosticEnvModeElement</a>	*	aggr	This aggregation contains a representation of Mode Declarations in the context of a DiagnosticEnvironmentalCondition.

**Table 5.9: DiagnosticEnvironmentalCondition**

### 5.4.1 Environmental Condition Formula

The core part of a [DiagnosticEnvironmentalCondition](#) is the [DiagnosticEnvConditionFormula](#).

Class	DiagnosticEnvConditionFormula			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
Note	<p>A <a href="#">DiagnosticEnvConditionFormula</a> embodies the computation instruction that is to be evaluated at runtime to determine if the <a href="#">DiagnosticEnvironmentalCondition</a> 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 <a href="#">DiagnosticEnvConditionFormula.op</a>.</p> <p>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 <a href="#">DiagnosticEnvConditionFormula.nrcValue</a>.</p>			
Base	<i>ARObject</i> , <a href="#">DiagnosticEnvConditionFormulaPart</a>			
Attribute	Type	Mul.	Kind	Note
nrcValue	PositiveInteger	0..1	attr	This attribute represents the concrete NRC value that shall be returned if the condition fails.
op	<a href="#">DiagnosticLogicalOperatorEnum</a>	1	attr	This attribute represents the concrete operator (supported operators: and, or) of the condition formula.
part (ordered)	<a href="#">DiagnosticEnvConditionFormulaPart</a>	*	aggr	This aggregation represents the collection of formula parts that can be combined by logical operators.

**Table 5.10: DiagnosticEnvConditionFormula**

Enumeration	DiagnosticLogicalOperatorEnum
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition
Note	Logical AND and OR operation (&&,   )
Literal	Description
logicalAnd	Logical AND <b>Tags:</b> atp.EnumerationValue=0
logicalOr	Logical OR <b>Tags:</b> atp.EnumerationValue=1

**Table 5.11: DiagnosticLogicalOperatorEnum**

[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 `parts` which are combined by the logical operations represented by the attribute `op`. ]([RS\\_DEXT\\_00079](#))

<b>Class</b>	<code>DiagnosticEnvConditionFormulaPart</code> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
<b>Note</b>	A <code>DiagnosticEnvConditionFormulaPart</code> can either be a atomic condition, e.g. a <code>DiagnosticEnvCompareCondition</code> , or a <code>DiagnosticEnvConditionFormula</code> , again, which allows arbitrary nesting.			
<b>Base</b>	<code>ARObject</code>			
<b>Subclasses</b>	<code>DiagnosticEnvCompareCondition</code> , <code>DiagnosticEnvConditionFormula</code>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
-	-	-	-	-

**Table 5.12: DiagnosticEnvConditionFormulaPart**

[TPS\_DEXT\_01114] **DiagnosticEnvConditionFormula that has no parts** [ A `DiagnosticEnvConditionFormula` that has no `parts` shall be evaluated to false. This rule shall apply independently of the value of `DiagnosticEnvConditionFormula.op`. ]([RS\\_DEXT\\_00079](#))

[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`. ]([RS\\_DEXT\\_00079](#))

[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 `parts` 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 `parts` 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 `parts` (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 `parts` 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 `parts` 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 SWS Dcm [10].

]([RS\\_DEXT\\_00079](#))

**[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 SWS Dcm [10] 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`. ]([RS\\_DEXT\\_00079](#))

**[constr\_1464] Allowed value range of `DiagnosticEnvConditionFormula.nrcValue`** [ 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 ISO 14229-1 [16].

## 5.4.2 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`.

<b>Class</b>	<code>DiagnosticEnvCompareCondition</code> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
<b>Note</b>	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 ( <code>==</code> , <code>!=</code> , <code>&lt;</code> , <code>&lt;=</code> , ...) is specified in <code>DiagnosticCompareCondition.compareType</code> .			
<b>Base</b>	<code>ARObject</code> , <code>DiagnosticEnvConditionFormulaPart</code>			
<b>Subclasses</b>	<code>DiagnosticEnvDataCondition</code> , <code>DiagnosticEnvModeCondition</code>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
<code>compareType</code>	<code>DiagnosticCompareTypeEnum</code>	1	attr	This attributes represents the concrete type of the comparison.

**Table 5.13: DiagnosticEnvCompareCondition**

<i>Enumeration</i>	<b>DiagnosticCompareTypeEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition
<b>Note</b>	Enumeration for the type of a comparison of values usually expressed by the following operators: ==, !=, <, <=, >, >=
<b>Literal</b>	<b>Description</b>
isEqual	equal <b>Tags:</b> atp.EnumerationValue=0
isGreaterOrEqual	greater than or equal <b>Tags:</b> atp.EnumerationValue=5
isGreaterThan	greater than <b>Tags:</b> atp.EnumerationValue=4
isLessOrEqual	less than or equal <b>Tags:</b> atp.EnumerationValue=3
isLessThan	less than <b>Tags:</b> atp.EnumerationValue=2
isNotEqual	not equal <b>Tags:</b> atp.EnumerationValue=1

**Table 5.14: DiagnosticCompareTypeEnum**

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

The specific sub-classes (the details are explained in sections 5.4.2.1 and 5.4.2.2) of [DiagnosticEnvCompareCondition](#) support a different set of attributes of [DiagnosticCompareTypeEnum](#) for setting the value of attribute `compareType`.

### 5.4.2.1 Data Condition

**[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`. ]([RS\\_DEXT\\_00079](#))

**[constr\_1465] 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`. ]()

<b>Class</b>	<b>DiagnosticEnvDataCondition</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
<b>Note</b>	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.			
<b>Base</b>	ARObject, <a href="#">DiagnosticEnvCompareCondition</a> , <a href="#">DiagnosticEnvConditionFormulaPart</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
compareValue	<a href="#">ValueSpecification</a>	1	aggr	This attribute represents a fixed compare value taken to evaluate the compare condition.
dataElement	<a href="#">DiagnosticDataElement</a>	1	ref	This reference represents the related diagnostic data element.

**Table 5.15: DiagnosticEnvDataCondition**

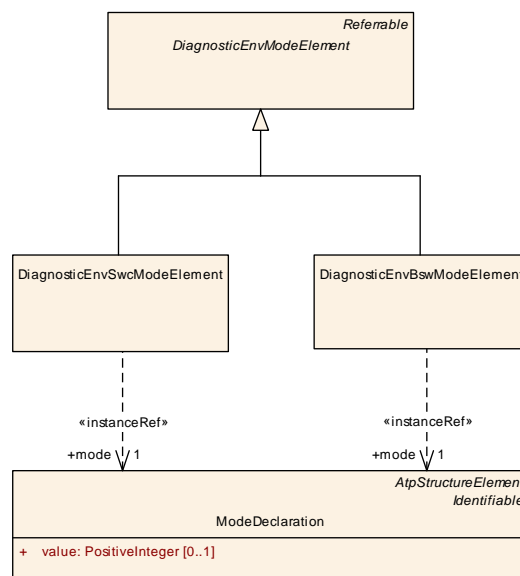
### 5.4.2.2 Mode Condition

[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. ] ([RS\\_DEXT\\_00079](#))

The concrete modeling of this aspect is sketched in Figure D.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 D.4) is that the [ModeDeclaration](#) can only be a member of the specific [ModeDeclarationGroup](#) taken to type the respective [ModeDeclarationGroupPrototype](#) anyway.



**Figure 5.4: Specializations of [DiagnosticEnvModeElement](#)**

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].

**[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`.

](*RS\_DEXT\_00079*)

Please note that the two alternatives mentioned in [TPS\_DEXT\_01120] are described in Figure 5.4.

**[constr\_1466] Allowed values of `compareType` in the context of a `DiagnosticEnvModeCondition`** [ 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`** [ 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`.



<b>Class</b>	<b>DiagnosticEnvModeCondition</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
<b>Note</b>	<p>DiagnosticEnvModeCondition are atomic condition based on the comparison of the active Mode Declaration in a ModeDeclarationGroupPrototype with the constant value of a ModeDeclaration.</p> <p>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.</p> <p>Only DiagnosticCompareTypeEnum.isEqual or DiagnosticCompareTypeEnum.isNotEqual are eligible values for DiagnosticAtomicCondition.compareType.</p>			
<b>Base</b>	ARObject, <a href="#">DiagnosticEnvCompareCondition</a> , <a href="#">DiagnosticEnvConditionFormulaPart</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
modeElement	<a href="#">DiagnosticEnvModeElement</a>	1	ref	This reference represents both the ModeDeclaration GroupPrototype and the ModeDeclaration relevant for the mode comparison.

**Table 5.16: DiagnosticEnvModeCondition**

<b>Class</b>	<b>DiagnosticEnvModeElement</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
<b>Note</b>	<p>All ModeDeclarations that are referenced in a DiagnosticEnvModeCondition must be defined as a DiagnosticEnvModeElement of this DiagnosticEnvironmentalCondition.</p> <p>This concept keeps the ARXML clean: It avoids that the DiagnosticEnvConditionFormula is cluttered by lengthy InstanceRef definitions.</p> <p>Furthermore, it allows that an InstanceRef only needs to be defined once and can be used multiple times in the different DiagnosticEnvModeConditions.</p>			
<b>Base</b>	ARObject, <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">DiagnosticEnvBswModeElement</a> , <a href="#">DiagnosticEnvSwcModeElement</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 5.17: DiagnosticEnvModeElement**

<b>Class</b>	<b>DiagnosticEnvSwcModeElement</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
<b>Note</b>	This meta-class represents the ability to refer to a ModeDeclaration in a concrete System context.			
<b>Base</b>	ARObject, <a href="#">DiagnosticEnvModeElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
mode	<a href="#">ModeDeclaration</a>	1	iref	This reference identifies both the ModeDeclarationGroup Prototype and the ModeDeclaration for the specific mode comparison.

**Table 5.18: DiagnosticEnvSwcModeElement**

<b>Class</b>	<b>DiagnosticEnvBswModeElement</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
<b>Note</b>	This meta-class represents the ability to refer to a specific ModeDeclaration in the scope of a BswModule Description.			
<b>Base</b>	ARObject, <a href="#">DiagnosticEnvModeElement</a> , <a href="#">Referrable</a>			





Class		DiagnosticEnvBswModeElement		
Attribute	Type	Mul.	Kind	Note
mode	<a href="#">ModeDeclaration</a>	1	iref	This reference identifies both the ModeDeclarationGroup Prototype and the ModeDeclaration for the specific mode comparison.

**Table 5.19: DiagnosticEnvBswModeElement**

## 5.5 Diagnostic Services supported by AUTOSAR

The following sub-chapters describe the modeling of the collection of relevant diagnostic services as defined in the ISO 14229-1 [16]. This means that the definition of the AUTOSAR [DiagnosticExtract](#) does not explicitly support the total collection of diagnostic services as defined by [16].

Some of the 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\_01045] Supported diagnostic services** [ The table [5.20](#) shows the UDS services supported by the [DiagnosticExtract](#). ] ([RS\\_DEXT\\_00003](#), [RS\\_DEXT\\_00004](#), [RS\\_DEXT\\_00005](#), [RS\\_DEXT\\_00006](#), [RS\\_DEXT\\_00007](#), [RS\\_DEXT\\_00008](#), [RS\\_DEXT\\_00009](#), [RS\\_DEXT\\_00010](#), [RS\\_DEXT\\_00011](#), [RS\\_DEXT\\_00012](#), [RS\\_DEXT\\_00013](#), [RS\\_DEXT\\_00014](#), [RS\\_DEXT\\_00015](#), [RS\\_DEXT\\_00016](#), [RS\\_DEXT\\_00017](#), [RS\\_DEXT\\_00018](#), [RS\\_DEXT\\_00019](#), [RS\\_DEXT\\_00020](#), [RS\\_DEXT\\_00021](#), [RS\\_DEXT\\_00022](#))

ID	Service
0x10	SessionControl
0x11	EcuReset
0x14	ClearDiagnosticInformation
0x19	ClearDTCInformation
0x22	ReadDataByIdentifier
0x23	ReadMemoryByAddress
0x27	SecurityAccess
0x28	CommunicationControl
0x2A	ReadDataByPeriodicIdentifier
0x2C	DynamicallyDefineDataIdentifier
0x2E	WriteDataByIdentifier
0x2F	IOControl
0x31	RoutineControl
0x34	RequestDownload
0x35	RequestUpload





ID	Service
0x36	TransferData
0x37	RequestTransferExit
0x3D	WriteMemoryByAddress
0x85	ControlDTCSetting
0x86	ResponseOnEvent

**Table 5.20: Supported diagnostic services**

**[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 ISO 14229-1 [16], 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. ]  
([RS\\_DEXT\\_00049](#), [RS\\_DEXT\\_00051](#))

**[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. ]([RS\\_DEXT\\_00001](#), [RS\\_DEXT\\_00051](#))

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 in order to avoid collisions that could occur if or when the list of possible values claimed by the AUTOSAR standard itself is extended.  
Example:

**Listing 5.2: Example for the definition of a custom `category`**

```
<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>
```

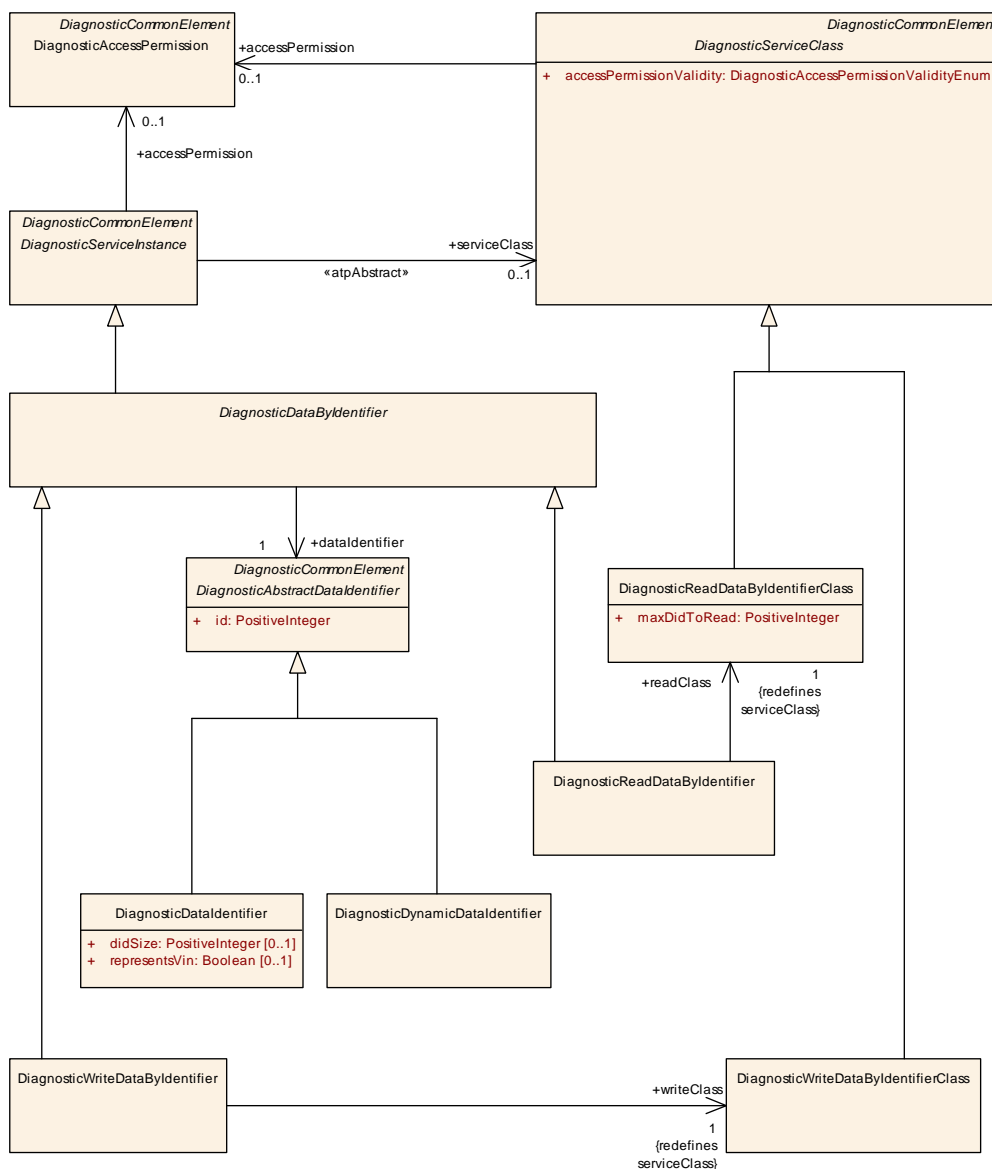
### 5.5.1 DataByIdentifier

This chapter describes the 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 *DiagnosticDataIdentifier*s as well as 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*.



**Figure 5.5: Modeling of diagnostic services ReadDataByIdentifier (0x22) and WriteDataByIdentifier (0x2E)**

[TPS\_DEXT\_01054] Existence of [DiagnosticDataByIdentifier.dataIdentifier](#) [ The configuration of a given [DiagnosticDataByIdentifier](#) is considered incomplete until the reference in the role [DiagnosticDataByIdentifier.dataIdentifier](#) exists. ]([RS\\_DEXT\\_0007](#), [RS\\_DEXT\\_00013](#), [RS\\_DEXT\\_00034](#))

The meaning of [TPS\_DEXT\_01054] is that the reference may be missing in intermediate steps of the configuration work flow. But at the point in time where ECU configuration is generated from the [DiagnosticExtract](#) the reference is needed to be able to make sense of the model for the given [DiagnosticDataByIdentifier](#).

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](#).

<b>Class</b>	<b>DiagnosticReadDataByIdentifier</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier			
<b>Note</b>	This represents an instance of the "Read Data by Identifier" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticDataByIdentifiers			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticDataByIdentifier</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
readClass	<a href="#">DiagnosticReadDataByIdentifierClass</a>	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 <a href="#">DiagnosticReadDataByIdentifier</a> in the given context.

**Table 5.21: DiagnosticReadDataByIdentifier**

<b>Class</b>	<b>DiagnosticWriteDataByIdentifier</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier			
<b>Note</b>	This represents an instance of the "Write Data by Identifier" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticDataByIdentifiers			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticDataByIdentifier</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
writeClass	<a href="#">DiagnosticWriteDataByIdentifierClass</a>	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 <a href="#">DiagnosticWriteDataByIdentifier</a> in the given context.

**Table 5.22: DiagnosticWriteDataByIdentifier**

<b>Class</b>	<b>DiagnosticWriteDataByIdentifierClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Write Data by Identifier" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticDataByIdentifiers			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
-	-	-	-	-

**Table 5.23: DiagnosticWriteDataByIdentifierClass**

<b>Class</b>	<b>DiagnosticDataByIdentifier</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier			
<b>Note</b>	This represents an abstract base class for all diagnostic services that access data by identifier.			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">DiagnosticReadDataByIdentifier</a> , <a href="#">DiagnosticReadScalingDataByIdentifier</a> , <a href="#">DiagnosticWriteDataByIdentifier</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataIdentifier	<a href="#">DiagnosticAbstractDataIdentifier</a>	1	ref	This represents the linked DiagnosticDataIdentifier.

**Table 5.24: DiagnosticDataByIdentifier**

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.

<b>Class</b>	<b>DiagnosticReadDataByIdentifierClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Read Data by Identifier" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticDataByIdentifiers			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
maxDidToRead	PositiveInteger	1	attr	This attribute represents the maximum number of allowed DIDs in a single instance of DiagnosticReadDataByIdentifier.

**Table 5.25: DiagnosticReadDataByIdentifierClass**

Please note that it is possible to create a reference to a concrete [DiagnosticDataIdentifier](#) from different [DiagnosticServiceInstances](#).

**[TPS\_DEXT\_01050] Consistency of [DiagnosticServiceSwMapping](#) with respect to data IDs** [ For each [DiagnosticServiceSwMapping](#) that references a [DiagnosticValueNeeds](#) and a [DiagnosticDataByIdentifier](#), the value

of `DiagnosticValueNeeds.didNumber` shall be ignored and the value of `DiagnosticDataByIdentifier.dataIdentifier.id` shall be taken instead. ]  
([RS\\_DEXT\\_00007](#), [RS\\_DEXT\\_00013](#), [RS\\_DEXT\\_00034](#), [RS\\_DEXT\\_00052](#))

<b>Class</b>	<b>DiagnosticValueNeeds</b>			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
<b>Note</b>	<p>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.</p> <p>In the case of using a sender receiver communicated value, the related value shall be taken via assigned Data in the role "signalBasedDiagnostics".</p> <p>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).</p>			
<b>Base</b>	<i>ARObject</i> , <i>DiagnosticCapabilityElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> , <i>ServiceNeeds</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataLength	PositiveInteger	0..1	attr	This attribute is applicable only if the ServiceNeed is aggregated within BswModuleDependency. This attribute represents the length of data (in bytes) provided for this particular PID signal.
diagnosticValueAccess	DiagnosticValueAccessEnum	0..1	attr	This attribute controls whether the data can be read and written or whether it is to be handled read-only.
didNumber	PositiveInteger	0..1	attr	This represents a Data identifier for the diagnostic value. This allows to predefine the DID number if the responsible function developer has received a particular requirement from the OEM or from a standardization body.
fixedLength	Boolean	0..1	attr	This attribute controls whether the data length of the data is fixed.
processingStyle	DiagnosticProcessingStyleEnum	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 5.26: DiagnosticValueNeeds**

[[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. ]([RS\\_DEXT\\_00007](#), [RS\\_DEXT\\_00034](#))

[[constr\\_1623](#)] **Restriction on DiagnosticReadScalingDataByIdentifier.dataIdentifier** [ The reference `DiagnosticReadScalingDataByIdentifier.dataIdentifier` shall only refer to a `DiagnosticDataIdentifier`. ]()

[[constr\\_1624](#)] **Existence of DiagnosticDataElement.scalingInfoSize** [ The attribute `DiagnosticDataElement.scalingInfoSize` shall only exist if the enclosing `DiagnosticParameter` is aggregated by a `DiagnosticDataIdentifier` that is referenced by a `DiagnosticReadScalingDataByIdentifier` in the role `DiagnosticReadScalingDataByIdentifier.dataIdentifier`. ]()

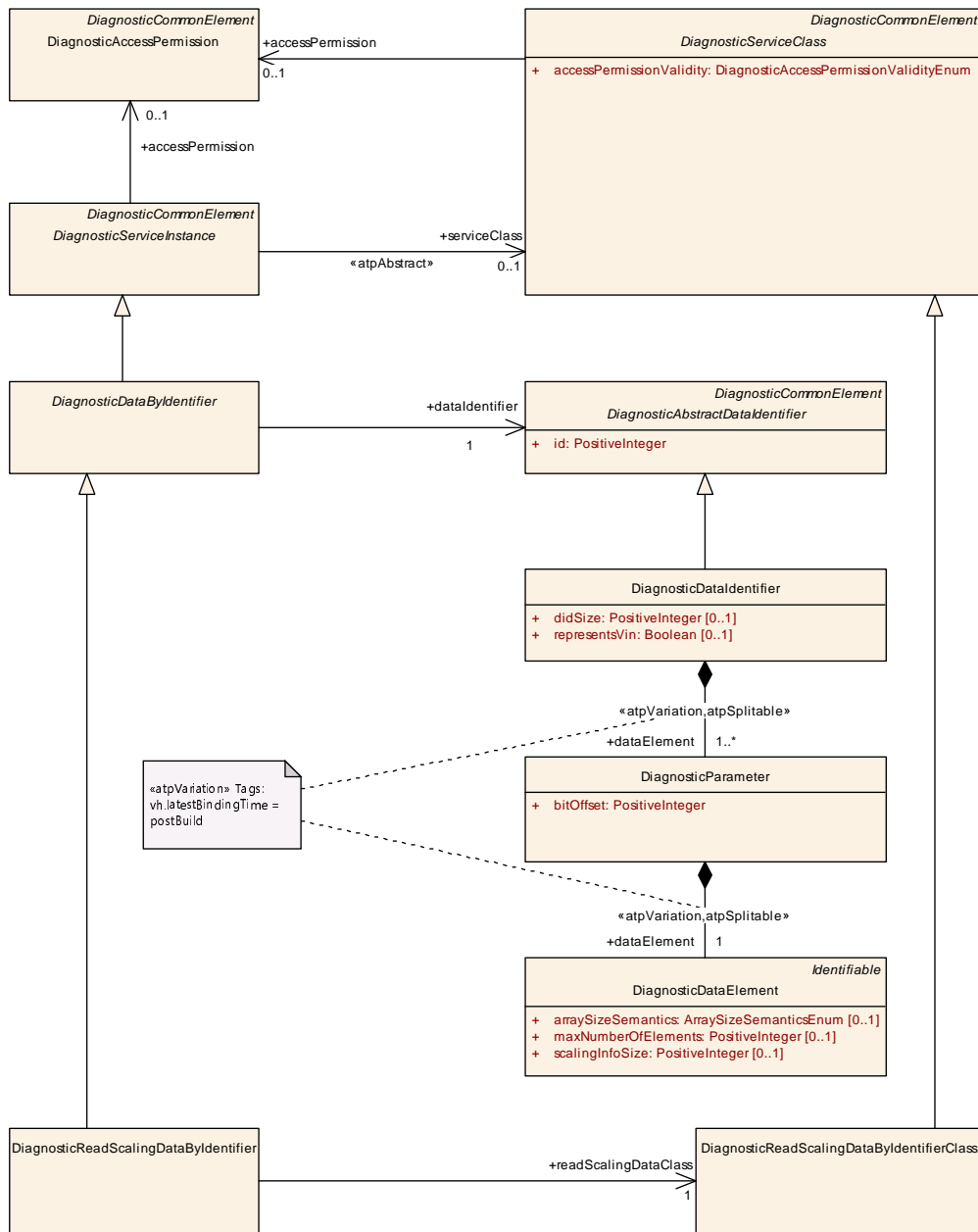


Figure 5.6: Modeling of diagnostic services `ReadScalingDataByIdentifier` (0x24)



<b>Class</b>	<b>DiagnosticReadScalingDataByIdentifier</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier			
<b>Note</b>	This represents an instance of the "Read Scaling Data by Identifier" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticDataByIdentifiers			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticDataByIdentifier</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
readScalingDataClass	<a href="#">DiagnosticReadScalingDataByIdentifierClass</a>	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 DiagnosticReadScalingDataByIdentifier in the given context.

**Table 5.27: DiagnosticReadScalingDataByIdentifier**

<b>Class</b>	<b>DiagnosticReadScalingDataByIdentifierClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Read Scaling Data by Identifier" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticDataByIdentifiers			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 5.28: DiagnosticReadScalingDataByIdentifierClass**

## 5.5.2 IOControl

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

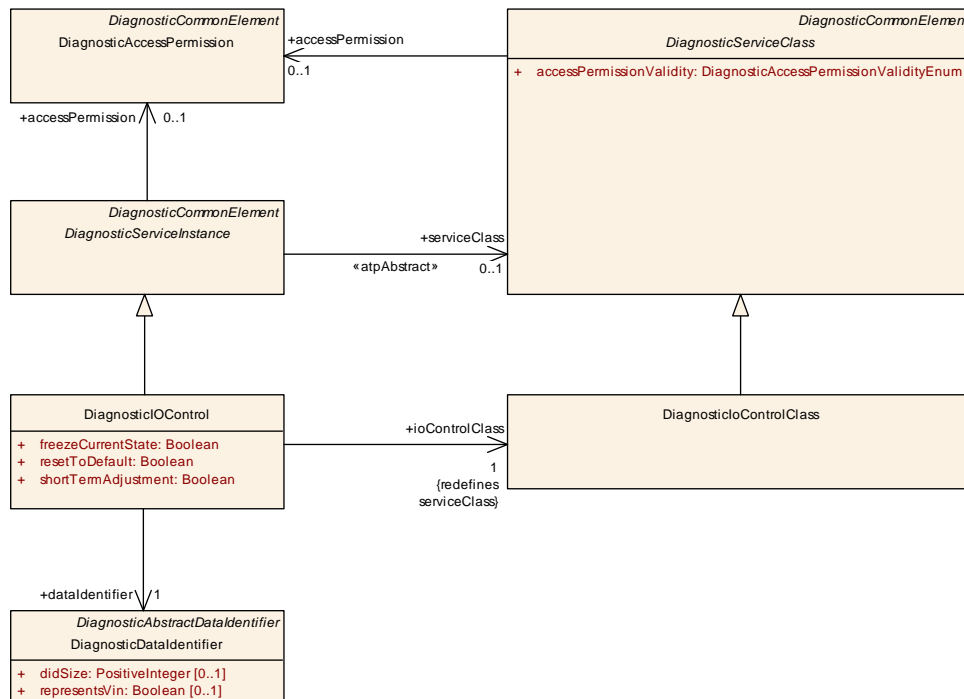


Figure 5.7: Modeling of diagnostic service IOControl (0x2F)

[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. ]  
(RS\_DEXT\_00014)

[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. ]  
(RS\_DEXT\_00014)

[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. ](RS\_DEXT\_00014, RS\_DEXT\_00034)

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

At run-time, only some `DiagnosticDataElements` 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**. ]([RS\\_DEXT\\_00037](#))

<b>Class</b>	<b>DiagnosticIOControl</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::IOControl			
<b>Note</b>	This represents an instance of the "I/O Control" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticIoControls			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataIdentifier	<a href="#">DiagnosticDataIdentifier</a>	1	ref	This represents the corresponding DiagnosticData Identifier
freezeCurrentState	Boolean	1	attr	Setting this attribute to true represents the ability of the Dcm to execute a freezeCurrentState.
ioControlClass	<a href="#">DiagnosticIoControlClass</a>	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.
resetToDefault	Boolean	1	attr	Setting this attribute to true represents the ability of the Dcm to execute a resetToDefault.
shortTermAdjustment	Boolean	1	attr	Setting this attribute to true represents the ability of the Dcm to execute a shortTermAdjustment.

**Table 5.29: DiagnosticIOControl**

<b>Class</b>	<b>DiagnosticIoControlClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::IOControl			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "IO Control" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticIoControls			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
—	—	—	—	—

**Table 5.30: 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. ]([RS\\_DEXT\\_00014](#), [RS\\_DEXT\\_00051](#))

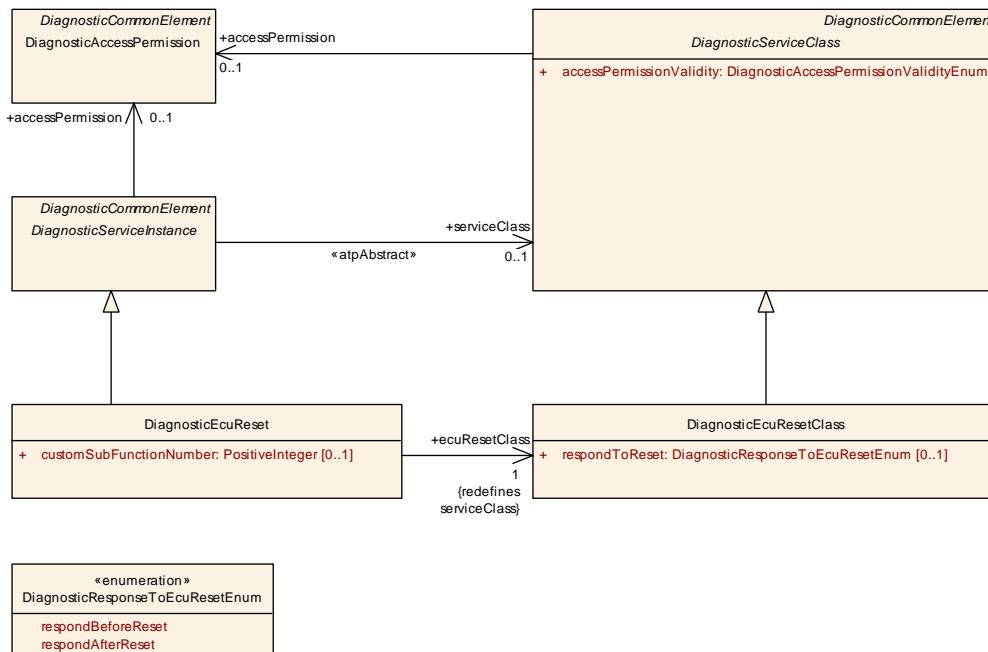
[TPS\_DEXT\_01051] **Consistency of DiagnosticServiceSwMapping with respect to data IDs** [ For each **DiagnosticServiceSwMapping** that references a **DiagnosticIoControlNeeds** and a **DiagnosticIOControl**, the value of **DiagnosticIoControlNeeds.didNumber** shall be ignored and the value of **DiagnosticIOControl.dataIdentifier.id** shall be taken instead. ]([RS\\_DEXT\\_00014](#), [RS\\_DEXT\\_00052](#))

<b>Class</b>	<b>DiagnosticControlNeeds</b>			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
<b>Note</b>	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.			
<b>Base</b>	ARObject, DiagnosticCapabilityElement, Identifiable, MultilanguageReferrable, Referrable, Service Needs			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
currentValue	DiagnosticValueNeeds	0..1	ref	Reference to the DiagnosticValueNeeds indicating the access to the current value via signalBasedDiagnostics.
didNumber	PositiveInteger	0..1	attr	This represents a Data identifier for the diagnostic value. This allows to predefine the DID number if the a function developer has received a particular requirement from the OEM or from a standardization body.
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 5.31: DiagnosticControlNeeds**

### 5.5.3 EcuReset

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



**Figure 5.8: Modeling of diagnostic service EcuReset (0x11)**

<b>Class</b>	<b>DiagnosticEcuReset</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EcuReset			
<b>Note</b>	This represents an instance of the "ECU Reset" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticEcuResets			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
customSubFunctionNumber	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	<a href="#">DiagnosticEcuResetClass</a>	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.

**Table 5.32: DiagnosticEcuReset**

<b>Class</b>	<b>DiagnosticEcuResetClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EcuReset			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Ecu Reset" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticEcuResets			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
respondToReset	<a href="#">DiagnosticResponseToEcuResetEnum</a>	0..1	attr	This attribute defines whether the response to the Ecu Reset service shall be transmitted before or after the actual reset.

**Table 5.33: DiagnosticEcuResetClass**

<b>Enumeration</b>	<b>DiagnosticResponseToEcuResetEnum</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EcuReset			
<b>Note</b>				
<b>Literal</b>	<b>Description</b>			
respondAfterReset	Answer to EcuReset service should come after the reset. <b>Tags:</b> atp.EnumerationValue=0			
respondBeforeReset	Answer to EcuReset service should come before the reset. <b>Tags:</b> atp.EnumerationValue=1			

**Table 5.34: DiagnosticResponseToEcuResetEnum**

Please note that (as already explained in section 5.5) 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 ISO document [16]. ]  
([RS\\_DEXT\\_00001](#), [RS\\_DEXT\\_00004](#), [RS\\_DEXT\\_00051](#))

**[TPS\_DEXT\_01019] Correspondence of `category` values to numerical values mentioned in the ISO 14229-1** [ The ISO 14229-1 [16] 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 ISO 14229-1 [16] standard document. ]([RS\\_DEXT\\_00001](#), [RS\\_DEXT\\_00004](#), [RS\\_DEXT\\_00051](#))

**[TPS\_DEXT\_01020] Manufacturer-specific values for sub-functions of service `EcuReset`** [ The ISO 14229-1 [16] 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]. ]([RS\\_DEXT\\_00001](#), [RS\\_DEXT\\_00004](#), [RS\\_DEXT\\_00051](#))

**[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.

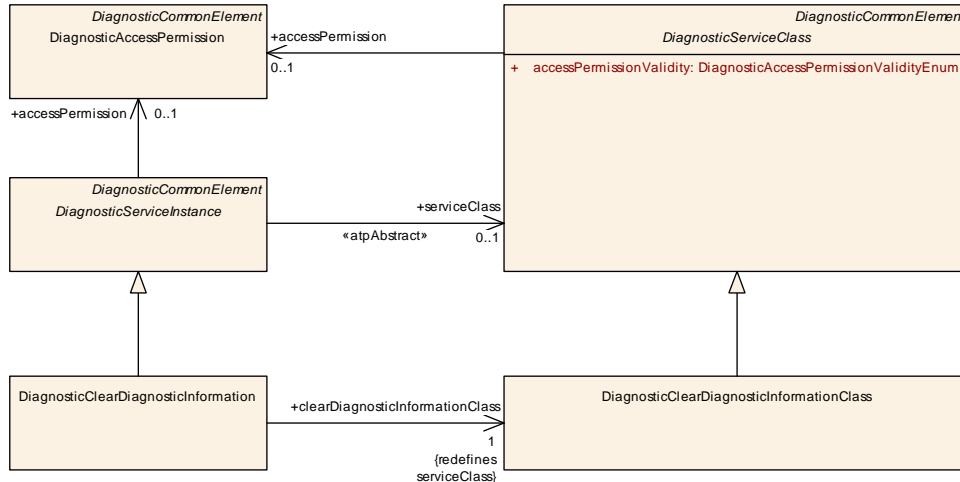
The tuple created by the the values of attributes `DiagnosticEcuReset.category` and `DiagnosticEcuReset.customSubFunctionNumber` fully specifies identification of the manufacturer- or supplier-specific sub-function. ]([RS\\_DEXT\\_00004](#), [RS\\_DEXT\\_00047](#), [RS\\_DEXT\\_00051](#))

**[constr\_1331] Existence of `DiagnosticEcuReset.customSubFunctionNumber`** [ 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.customSubFunctionNumber`** [ The allowed value for `DiagnosticEcuReset.customSubFunctionNumber` shall always be within the closed interval `0x40 .. 0x7E`. ]()

### 5.5.4 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.



**Figure 5.9: Modeling of diagnostic service `ClearDiagnosticInformation` (0x14)**

Please note that there is nothing to configure for `DiagnosticClearDiagnosticInformation` beyond its mere existence.

<b>Class</b>	<b>DiagnosticClearDiagnosticInformation</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ClearDiagnosticInfo			
<b>Note</b>	This represents an instance of the "Clear Diagnostic Information" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticClearDiagnosticInformations			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
clearDiagnosticInformationClass	<a href="#">DiagnosticClearDiagnosticInformationClass</a>	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 DiagnosticClearDiagnosticInformation in the given context.

**Table 5.35: DiagnosticClearDiagnosticInformation**

<b>Class</b>	<b>DiagnosticClearDiagnosticInformationClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ClearDiagnosticInfo			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Clear Diagnostic Information" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticClearDiagnosticInformations			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			





Class		DiagnosticClearDiagnosticInformationClass		
Attribute	Type	Mul.	Kind	Note
-	-	-	-	-

**Table 5.36: 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. ]([RS\\_DEXT\\_00001](#), [RS\\_DEXT\\_00005](#), [RS\\_DEXT\\_00051](#))

### 5.5.5 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.

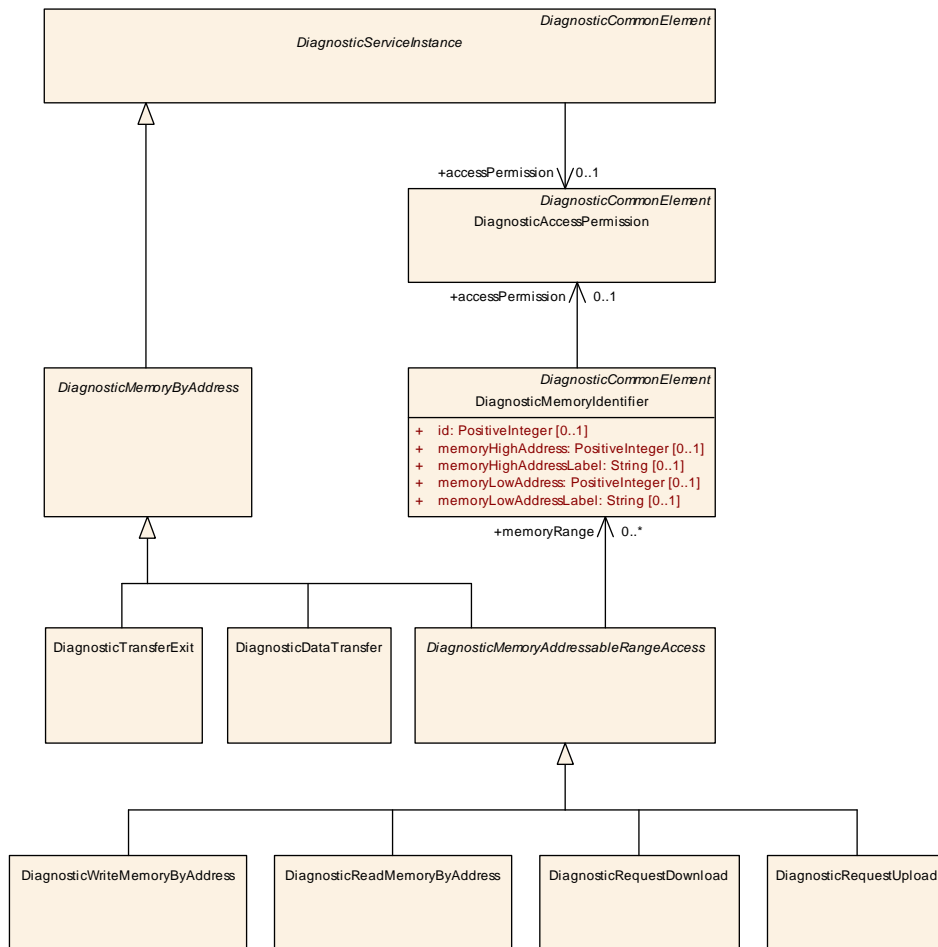
The service description for accessing memory for diagnostic purposes is modeled by the abstract meta-class `DiagnosticMemoryByAddress`. It is supposed to provide all model properties relevant for the memory access.

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`.

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 a documentation of the diagnostic configuration.





**Figure 5.10: Modeling of diagnostic services Memory ((0x23, 0x3D, 0x34-0x37)**

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 `DiagnosticMemoryIdentifier.memoryLowAddress` and `DiagnosticMemoryIdentifier.memoryHighAddress`** [ The attributes `DiagnosticMemoryIdentifier.memoryLowAddress` as well as `DiagnosticMemoryIdentifier.memoryHighAddress` shall not exist if the `DiagnosticMemoryIdentifier` referenced in the role `memoryRange` is referenced by a `DiagnosticRequestDownload` or a `DiagnosticRequestUpload`. ]()

**[constr\_1411] Existence of `DiagnosticMemoryIdentifier.memoryHighAddressLabel` vs. `DiagnosticMemoryIdentifier.memoryHighAddress`** [ 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](#)** [ 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](#)].

<b>Class</b>	<i>DiagnosticMemoryByAddress</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
<b>Note</b>	This represents an abstract base class for diagnostic services that deal with accessing memory by address.			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">DiagnosticDataTransfer</a> , <a href="#">DiagnosticMemoryAddressableRangeAccess</a> , <a href="#">DiagnosticTransferExit</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 5.37: DiagnosticMemoryByAddress**

<b>Class</b>	<i>DiagnosticMemoryAddressableRangeAccess</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
<b>Note</b>	This abstract base class			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMemoryByAddress</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">DiagnosticReadMemoryByAddress</a> , <a href="#">DiagnosticRequestDownload</a> , <a href="#">DiagnosticRequestUpload</a> , <a href="#">DiagnosticWriteMemoryByAddress</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
memoryRange	<a href="#">DiagnosticMemoryIdentifier</a>	*	ref	This represents the formal description of the memory segment to which the <a href="#">DiagnosticMemoryByAddress</a> applies.

**Table 5.38: DiagnosticMemoryAddressableRangeAccess**

<b>Class</b>	<b>DiagnosticMemoryIdentifier</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
<b>Note</b>	This meta-class represents the ability to define memory properties from the diagnostics point of view. <b>Tags:</b> atp.recommendedPackage=DiagnosticMemoryByAdresss			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
access Permission	<a href="#">DiagnosticAccessPermission</a>	0..1	ref	This represents that access permission defined for the specific DiagnosticMemoryIdentifier.
id	PositiveInteger	0..1	attr	This represents the identification of the memory segment.
memoryHigh Address	PositiveInteger	0..1	attr	This represents the upper bound for addresses of the memory segment.
memoryHigh AddressLabel	String	0..1	attr	This represents a symbolic label for the upper bound for addresses of the memory segment.
memoryLow Address	PositiveInteger	0..1	attr	This represents the lower bound for addresses of the memory segment.
memoryLow AddressLabel	String	0..1	attr	This represents a symbolic label for the lower bound for addresses of the memory segment.

**Table 5.39: DiagnosticMemoryIdentifier**

<b>Class</b>	<b>DiagnosticWriteMemoryByAddress</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
<b>Note</b>	This represents an instance of the "Write Memory by Address" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticMemoryByAdresss			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMemoryAddressableRangeAccess</a> , <a href="#">DiagnosticMemoryByAddress</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
writeClass	<a href="#">DiagnosticWriteMemoryByAddressClass</a>	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 DiagnosticWriteMemoryByAddress in the given context.

**Table 5.40: DiagnosticWriteMemoryByAddress**

<b>Class</b>	<b>DiagnosticWriteMemoryByAddressClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Write Memory by Address" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticMemoryByAdresss			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 5.41: DiagnosticWriteMemoryByAddressClass**

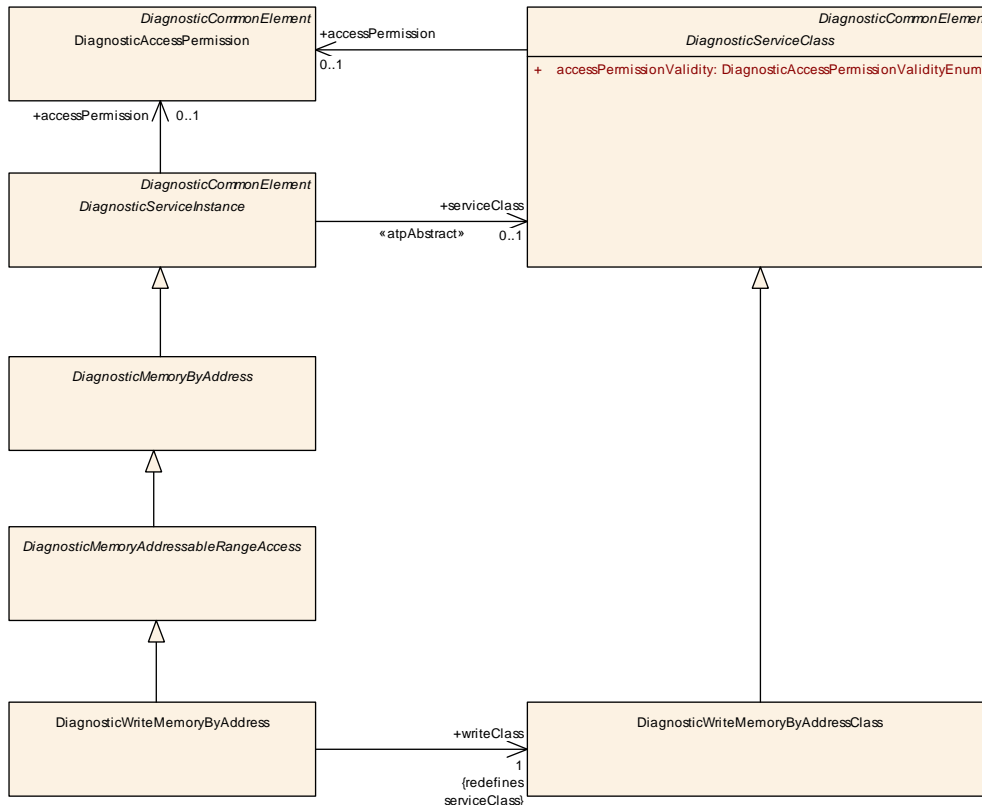


Figure 5.11: Modeling of diagnostic service WriteMemoryByAddress (0x3D)

[TPS\_DEXT\_01023] writeMemoryByAddress does not define any sub-functions

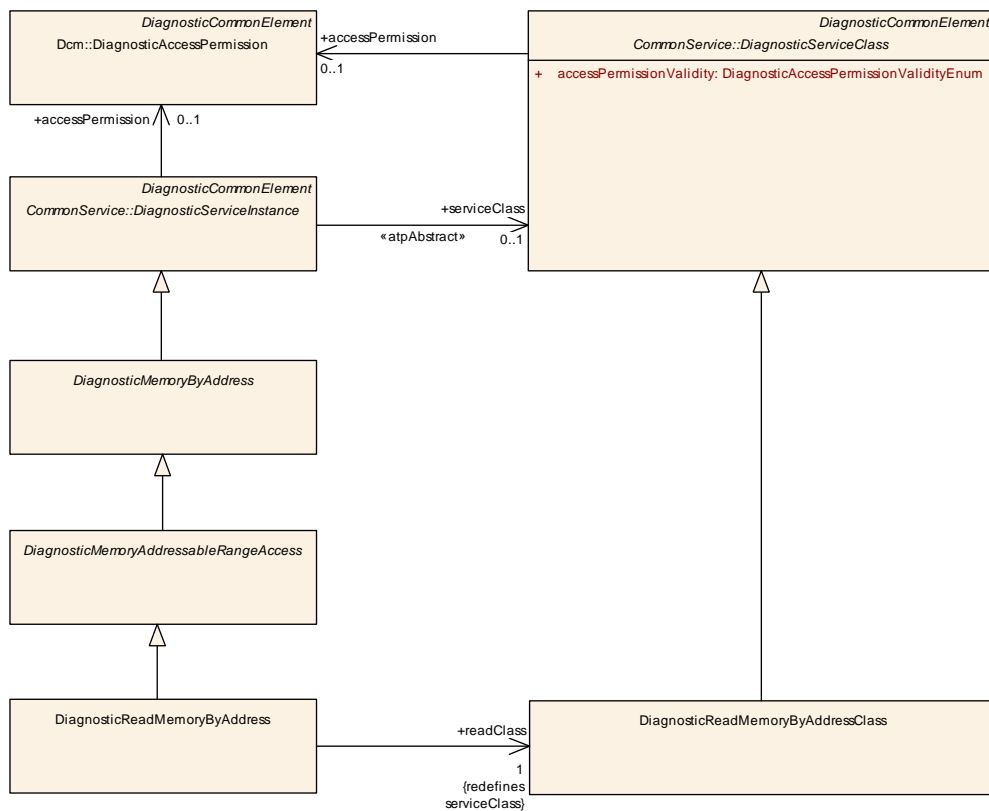
[ The diagnostic service WriteMemoryByAddress does not define any sub-functions, therefore the value of DiagnosticWriteMemoryByAddress.category does not need to be constrained. ] (RS\_DEXT\_0001, RS\_DEXT\_00020, RS\_DEXT\_00051)

<b>Class</b>	<b>DiagnosticReadMemoryByAddress</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
<b>Note</b>	This represents an instance of the "Read Memory by Address" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticMemoryByAdresss			
<b>Base</b>	ARElement, ARObjct, CollectableElement, DiagnosticCommonElement, DiagnosticMemoryAddressableRangeAccess, DiagnosticMemoryByAddress, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
readClass	DiagnosticReadMemoryByAddressClass	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 DiagnosticReadMemoryBy Adressst in the given context.

Table 5.42: DiagnosticReadMemoryByAddress

<b>Class</b>	<b>DiagnosticReadMemoryByAddressClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Read Memory by Address" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticMemoryByAdresss			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
-	-	-	-	-

**Table 5.43: DiagnosticReadMemoryByAddressClass**



**Figure 5.12: Modeling of diagnostic service ReadMemoryByAddress (0x23)**

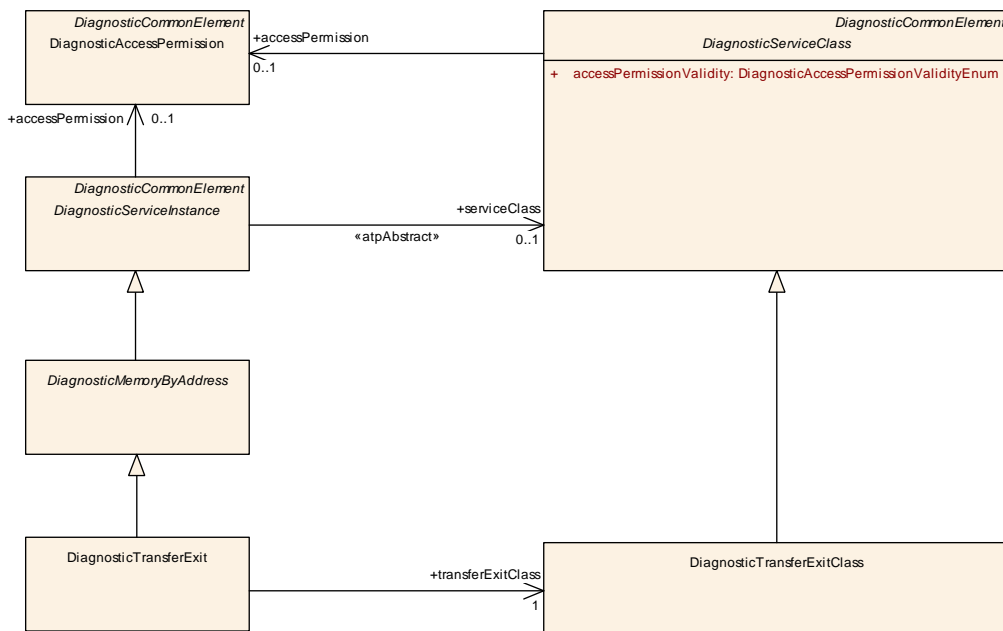
**[TPS\_DEXT\_01024] ReadMemoryByAddress does not define any sub-functions** [ The diagnostic service ReadMemoryByAddress does not define any sub-functions, therefore the value of `DiagnosticReadMemoryByAddress.category` does not need to be constrained. ] ([RS\\_DEXT\\_00001](#), [RS\\_DEXT\\_00008](#), [RS\\_DEXT\\_00051](#))

<b>Class</b>	<b>DiagnosticTransferExit</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
<b>Note</b>	This represents an instance of the "Transfer Exit" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticMemoryByAdresss			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMemoryByAddress</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
transferExit Class	<a href="#">DiagnosticTransferExit Class</a>	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.

**Table 5.44: DiagnosticTransferExit**

<b>Class</b>	<b>DiagnosticTransferExitClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Transfer Exit" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticMemoryByAdresss			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
—	—	—	—	—

**Table 5.45: DiagnosticTransferExitClass**



**Figure 5.13: Modeling of diagnostic service TransferExit (0x37)**

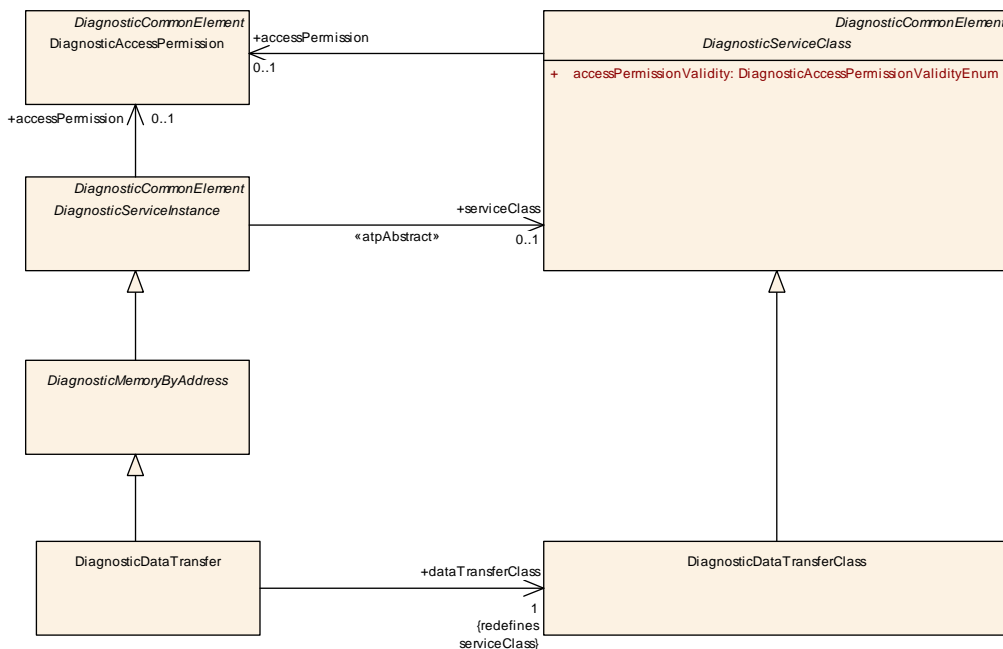
**[TPS\_DEXT\_01025] TransferExit does not define any sub-functions** [ The diagnostic service TransferExit does not define any sub-functions, therefore the value of `DiagnosticTransferExit.category` does not need to be constrained. ]([RS\\_DEXT\\_00001](#), [RS\\_DEXT\\_00019](#), [RS\\_DEXT\\_00051](#))

<b>Class</b>	<b>DiagnosticDataTransfer</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
<b>Note</b>	This represents an instance of the "Data Transfer" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticMemoryByAdresss			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMemoryByAddress</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataTransfer Class	<a href="#">DiagnosticDataTransfer Class</a>	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.

**Table 5.46: DiagnosticDataTransfer**

<b>Class</b>	<b>DiagnosticDataTransferClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Data Transfer" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticMemoryByAdresss			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
-	-	-	-	-

**Table 5.47: DiagnosticDataTransferClass**



**Figure 5.14: Modeling of diagnostic service DataTransfer (0x36)**

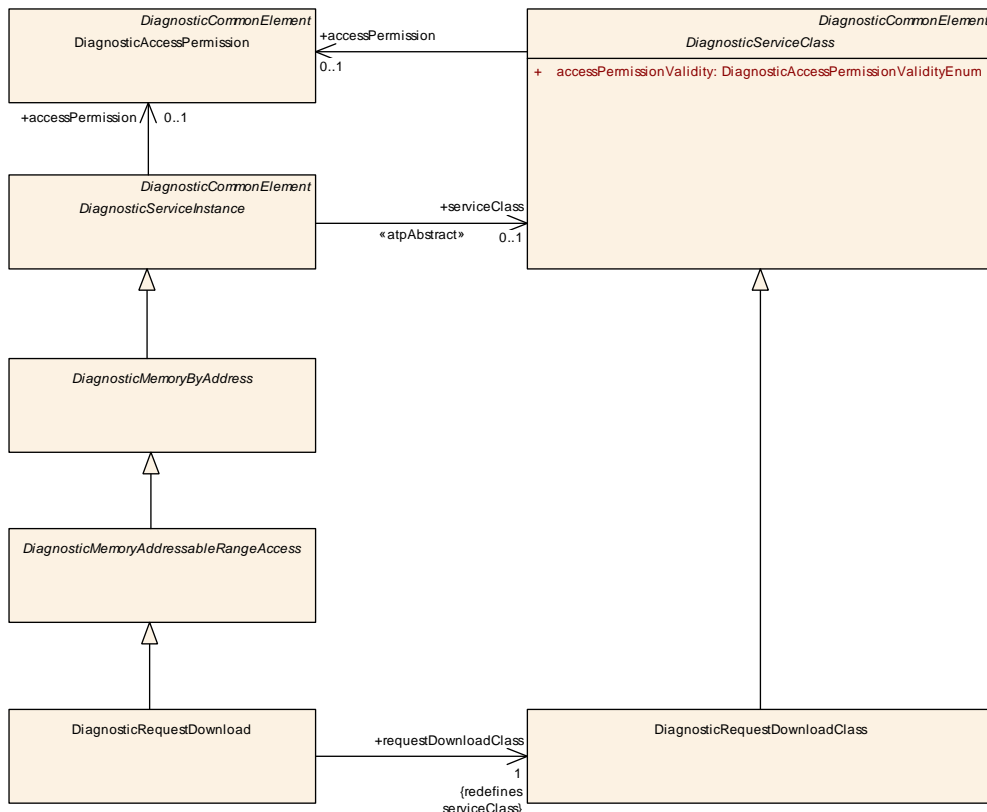
**[TPS\_DEXT\_01026] DataTransfer does not define any sub-functions** [ The diagnostic service DataTransfer does not define any sub-functions, therefore the value of `DiagnosticDataTransfer.category` does not need to be constrained. ]([RS\\_DEXT\\_0001](#), [RS\\_DEXT\\_00018](#), [RS\\_DEXT\\_00051](#))

<b>Class</b>	<b>DiagnosticRequestDownload</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
<b>Note</b>	This represents an instance of the "Request Download" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticMemoryByAdresss			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMemoryAddressableRangeAccess</a> , <a href="#">DiagnosticMemoryByAddress</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
request DownloadClass	<a href="#">DiagnosticRequestDownloadClass</a>	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.

**Table 5.48: DiagnosticRequestDownload**

<b>Class</b>	<b>DiagnosticRequestDownloadClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Request Download" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticMemoryByAdresss			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 5.49: DiagnosticRequestDownloadClass**



**Figure 5.15: Modeling of diagnostic service RequestDownLoad (0x34)**



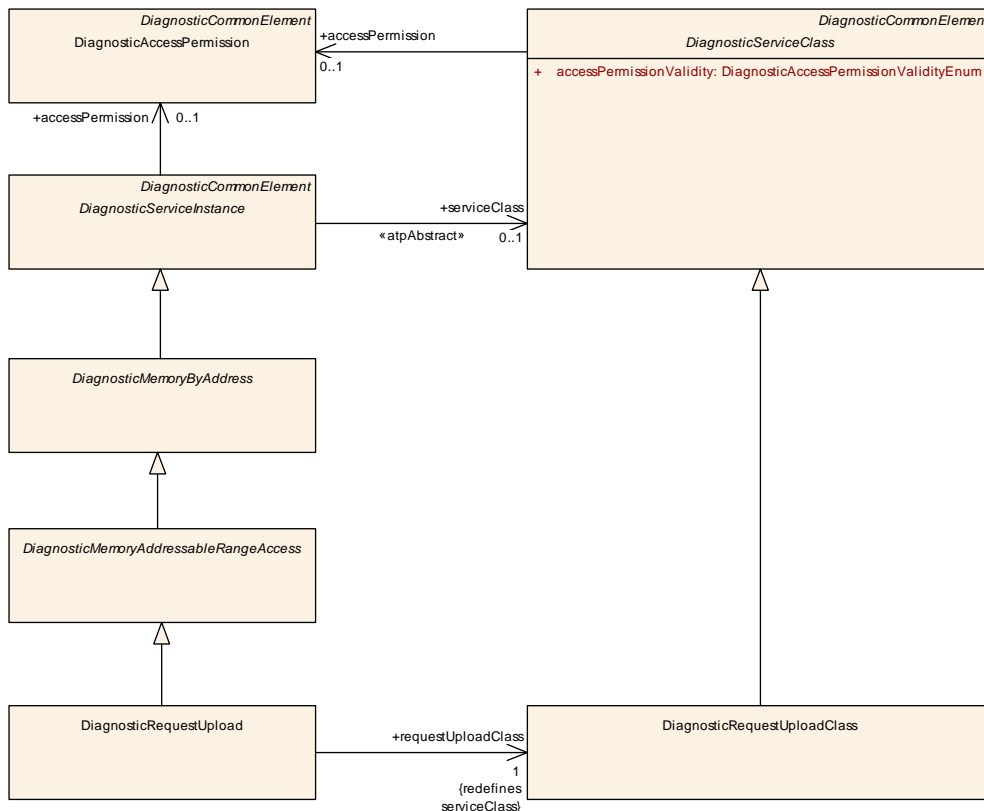
[TPS\_DEXT\_01027] **RequestDownload** does not define any sub-functions [ The diagnostic service `RequestDownload` does not define any sub-functions, therefore the value of `DiagnosticRequestDownload.category` does not need to be constrained. ] ([RS\\_DEXT\\_00001](#), [RS\\_DEXT\\_00016](#), [RS\\_DEXT\\_00051](#))

<b>Class</b>	<b>DiagnosticRequestUpload</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
<b>Note</b>	This represents an instance of the "Request Upload" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticMemoryByAdresss			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMemoryAddressableRangeAccess</a> , <a href="#">DiagnosticMemoryByAddress</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
requestUpload Class	<a href="#">DiagnosticRequestUploadClass</a>	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.

**Table 5.50: DiagnosticRequestUpload**

<b>Class</b>	<b>DiagnosticRequestUploadClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Request Upload" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticMemoryByAdresss			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
-	-	-	-	-

**Table 5.51: DiagnosticRequestUploadClass**



**Figure 5.16: Modeling of diagnostic service RequestDownload (0x35)**

**[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. ] ([RS\\_DEXT\\_0001](#), [RS\\_DEXT\\_00017](#), [RS\\_DEXT\\_00051](#))

### 5.5.6 CommunicationControl

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

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.

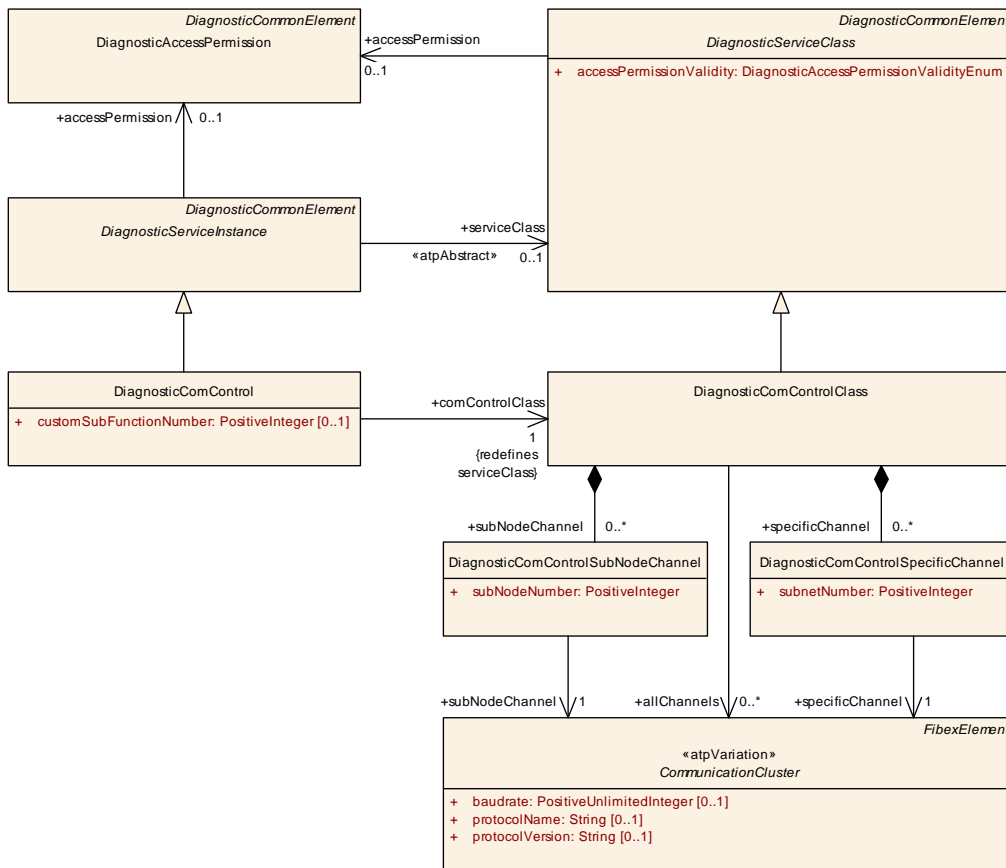


Figure 5.17: Modeling of diagnostic service CommunicationControl (0x28)

[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 ISO document [16]. ]  
([RS\\_DEXT\\_00001](#), [RS\\_DEXT\\_00010](#), [RS\\_DEXT\\_00051](#))

[TPS\_DEXT\_01029] Correspondence of **category** values to numerical values mentioned in the ISO 14229-1 [ The ISO 14229-1 [16] 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 ISO 14229-1 [16] standard document. `|(RS_DEXT_00001, RS_DEXT_00010, RS_DEXT_00051)`

**[TPS\_DEXT\_01030] Manufacturer-specific values for sub-functions of service `CommunicationControl`** [ The ISO 14229-1 [16] 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]. `|(RS_DEXT_00010, RS_DEXT_00047, RS_DEXT_00051)`

**[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. `|(RS_DEXT_00010, RS_DEXT_00047, RS_DEXT_00051)`

**[constr\_1334] Existence of `DiagnosticComControl.customSubFunctionNumber`** [ 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`** [ 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.

|(RS\_DEXT\_00010)

<b>Class</b>	«atpVariation» <b>CommunicationCluster</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
<b>Note</b>	<p>The CommunicationCluster is the main element to describe the topological connection of communicating ECUs.</p> <p>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.</p> <p>A CommunicationCluster aggregates one or more physical channels.</p> <p><b>Tags:</b> vh.latestBindingTime=postBuild</p>			
<b>Base</b>	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
<b>Subclasses</b>	AbstractCanCluster, EthernetCluster, FlexrayCluster, LinCluster, UserDefinedCluster			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
baudrate	PositiveUnlimitedInteger	0..1	attr	Channels speed in bits/s.
physicalChannel	PhysicalChannel	1..*	aggr	<p>This relationship defines which channel element belongs to which cluster. A channel must be assigned to exactly one cluster, whereas a cluster may have one or more channels.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=systemDesignTime</p>
protocolName	String	0..1	attr	The name of the protocol used.
protocolVersion	String	0..1	attr	The version of the protocol used.

**Table 5.52: CommunicationCluster**

[constr\_1336] Applicable value range for `DiagnosticComControlSpecificChannel.subnetNumber` [ The value of attribute `DiagnosticComControlSpecificChannel.subnetNumber` shall be within the closed interval **1 .. 14**. ]()

Please note that the regulation implied by [constr\_1336] has not been introduced on an arbitrary basis but gets its conceptual background from the ISO 14229-1 [16] 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](#) [ 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. ] ([RS\\_DEXT\\_00010](#))

<b>Class</b>	<b>DiagnosticComControl</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl			
<b>Note</b>	This represents an instance of the "Communication Control" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticCommunicationControls			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
comControlClass	<a href="#">DiagnosticComControlClass</a>	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.
customSubFunctionNumber	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 5.53: DiagnosticComControl**

<b>Class</b>	<b>DiagnosticComControlSpecificChannel</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl			
<b>Note</b>	This represents the ability to add further attributes to the definition of a specific channel that is subject to the diagnostic service "communication control".			
<b>Base</b>	<a href="#">ARObject</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
specificChannel	<a href="#">CommunicationCluster</a>	1	ref	This represents the affected CommunicationClusters in the role specificChannel
subnetNumber	PositiveInteger	1	attr	This represents the applicable subnet number (which is an arbitrary number ranging from 1..14)

**Table 5.54: DiagnosticComControlSpecificChannel**

<b>Class</b>	<b>DiagnosticComControlClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Communication Control" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticCommunicationControls			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
allChannels	<a href="#">CommunicationCluster</a>	*	ref	This reference represents the semantics that all available channels shall be affected. It is still necessary to refer to individual CommunicationClusters 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.
specificChannel	<a href="#">DiagnosticComControlSpecificChannel</a>	*	aggr	This represents the ability to add additional attributes to the case that only specific channels are supposed to be considered,
subNode Channel	<a href="#">DiagnosticComControlSubNodeChannel</a>	*	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".

**Table 5.55: DiagnosticComControlClass**

<b>Class</b>	<b>DiagnosticComControlSubNodeChannel</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl			
<b>Note</b>	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".			
<b>Base</b>	<a href="#">ARObject</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
subNode Channel	<a href="#">CommunicationCluster</a>	1	ref	This represents the affected CommunicationClusters in the role subNodeChannel
subNode Number	PositiveInteger	1	attr	This represents the applicable subNode number. The value corresponds to the request message parameter nodeIdentificationNumber of diagnostic service CommunicationControl (0x28).

**Table 5.56: DiagnosticComControlSubNodeChannel**

### 5.5.7 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`.

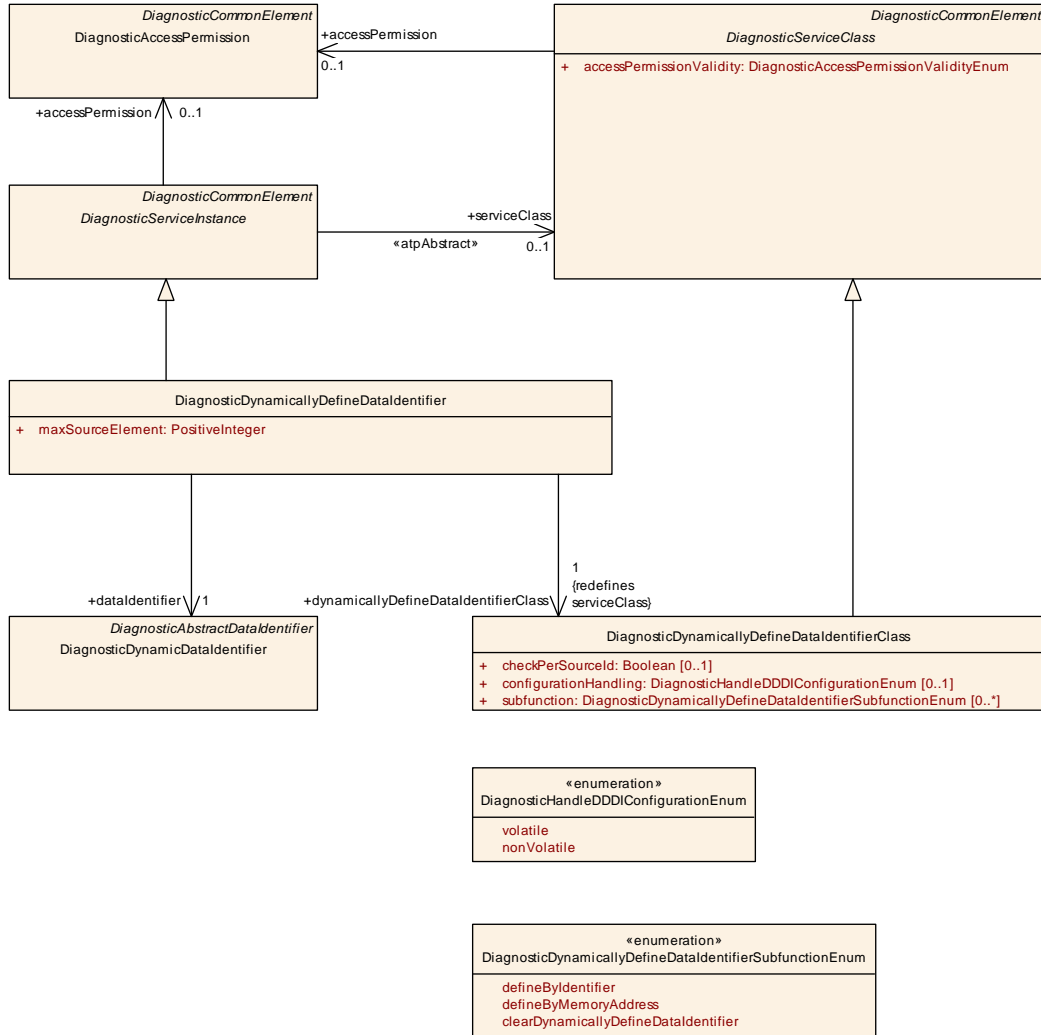


Figure 5.18: Modeling of diagnostic service `DynamicallyDefineDataIdentifier` (0x2C)

[constr\_1421] Consistency of `DiagnosticDynamicallyDefineDataIdentifierClass.subfunction` [ The values of `DiagnosticDynamicallyDefineDataIdentifierClass.subfunction` shall not repeat, i.e. every value of `DiagnosticDynamicallyDefineDataIdentifierSubfunctionEnum` shall at most appear once in the `subfunction` attribute. ]()



<b>Class</b>	<b>DiagnosticDynamicallyDefineDataIdentifier</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier			
<b>Note</b>	This represents an instance of the "Dynamically Define Data Identifier" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticDynamicallyDefineDataIdentifiers			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataIdentifier	<a href="#">DiagnosticDynamicDataIdentifier</a>	1	ref	This represents the applicable DiagnosticDynamicDataIdentifier.
dynamicallyDefineDataIdentifierClass	<a href="#">DiagnosticDynamicallyDefineDataIdentifierClass</a>	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 DiagnosticDynamicallyDefineDataIdentifier in the given context.
maxSourceElement	PositiveInteger	1	attr	This represents the maximum number of source elements of the dynamically created DID.

**Table 5.57: DiagnosticDynamicallyDefineDataIdentifier**

<b>Class</b>	<b>DiagnosticDynamicallyDefineDataIdentifierClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Dynamically Define Data Identifier" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticDynamicallyDefineDataIdentifiers			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
checkPerSourceId	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 0xF200 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.
configurationHandling	<a href="#">DiagnosticHandleDDDIConfigurationEnum</a>	0..1	attr	This configuration switch defines whether DDDID definition is handled as non-volatile information or not.
subfunction	<a href="#">DiagnosticDynamicallyDefineDataIdentifierSubfunctionEnum</a>	*	attr	This attribute contains a list of applicable subfunctions for all DiagnosticDynamicallyDefineDataIdentifier that reference the DiagnosticDynamicallyDefineDataIdentifierClass.

**Table 5.58: DiagnosticDynamicallyDefineDataIdentifierClass**

<b>Enumeration</b>	<b>DiagnosticHandleDDDIConfigurationEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier
<b>Note</b>	This meta-class represents the options for controlling how the configuration of the DynamicallyDefineDataIdentifiers is done in the given context.
<b>Literal</b>	<b>Description</b>
nonVolatile	This indicates that the configuration of DynamicallyDefineDataIdentifier shall be stored as non-volatile data. <b>Tags:</b> atp.EnumerationValue=0





<i>Enumeration</i>	<b>DiagnosticHandleDDDIConfigurationEnum</b>
volatile	This indicates that the configuration of DynamicallyDefineDataIdentifier shall be handled as volatile data. <b>Tags:</b> atp.EnumerationValue=1

**Table 5.59: DiagnosticHandleDDDIConfigurationEnum**

<i>Enumeration</i>	<b>DiagnosticDynamicallyDefineDataIdentifierSubfunctionEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier
<b>Note</b>	This meta-class contains a list of possible subfunctions for the UDS service 0x2C.
<b>Literal</b>	<b>Description</b>
clearDynamicallyDefineDataIdentifier	Clear the specified dynamic data identifier. <b>Tags:</b> atp.EnumerationValue=0
defineByIdentifier	The definition of dynamic data identifier shall be done via a reference to a diagnostic data identifier. <b>Tags:</b> atp.EnumerationValue=1
defineByMemoryAddress	The definition of dynamic data identifier shall be done via a reference to a memory address. <b>Tags:</b> atp.EnumerationValue=2

**Table 5.60: DiagnosticDynamicallyDefineDataIdentifierSubfunctionEnum**

### 5.5.8 ReadDataByPeriodicIdentifier

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



<b>Class</b>	<b>DiagnosticReadDataByPeriodicIDClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Read Data by periodic Identifier" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticReadDataByPeriodicIds			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
maxPeriodicDid ToRead	PositiveInteger	1	attr	This represents the maximum number of data identifiers that can be included in one request.
periodicRate	<a href="#">DiagnosticPeriodicRate</a>	1..*	aggr	This represents the description of a collection of periodic rates in which the service can be executed.
schedulerMax Number	PositiveInteger	1	attr	This represents the maximum number of periodic data identifiers that can be scheduled in parallel.

**Table 5.62: DiagnosticReadDataByPeriodicIDClass**

<b>Class</b>	<b>DiagnosticPeriodicRate</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID			
<b>Note</b>	This represents the ability to define a periodic rate for the specification of the "read data by periodic ID" diagnostic service.			
<b>Base</b>	<a href="#">ARObject</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
period	TimeValue	1	attr	This represents the period of the <a href="#">DiagnosticPeriodicRate</a> in seconds.
periodicRate Category	<a href="#">DiagnosticPeriodicRate CategoryEnum</a>	0..1	attr	This attribute represents the category of the periodic rate.

**Table 5.63: DiagnosticPeriodicRate**

<b>Enumeration</b>	<b>DiagnosticPeriodicRateCategoryEnum</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID			
<b>Note</b>	This meta-class provides possible values for the setting of the periodic rate.			
<b>Literal</b>	<b>Description</b>			
periodicRateFast	This value represents a fast periodic rate. <b>Tags:</b> atp.EnumerationValue=0			
periodicRate Medium	This value represents a medium periodic rate. <b>Tags:</b> atp.EnumerationValue=1			
periodicRateSlow	This value represents a slow periodic rate. <b>Tags:</b> atp.EnumerationValue=2			

**Table 5.64: DiagnosticPeriodicRateCategoryEnum**

The meaning of the values defined in [DiagnosticPeriodicRateCategoryEnum](#) is described in the applicable ISO document [16].

### 5.5.9 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.

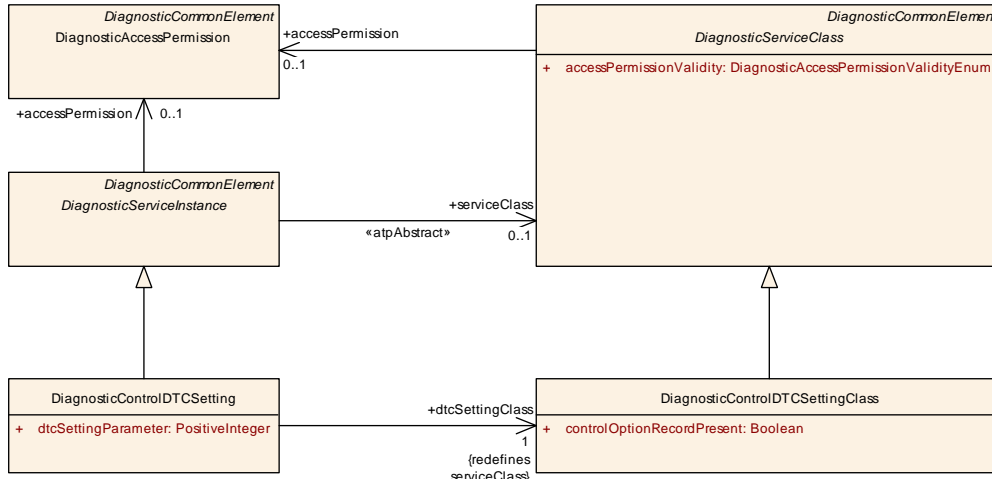


Figure 5.20: Modeling of diagnostic service `ControlDTCSetting` (0x85)

[TPS\_DEXT\_01075] standardized values for the attribute `DiagnosticControlDTCSetting.category` [ AUTOSAR does not standardize any of the possible values for the attribute `DiagnosticControlDTCSetting.category`. ]  
([RS\\_DEXT\\_00001](#), [RS\\_DEXT\\_00021](#), [RS\\_DEXT\\_00051](#))

[TPS\_DEXT\_01076] Identification of sub-functions of diagnostic service `ControlDTCSetting` [ The identification of sub-functions (for which the attribute `category` is used for several other `DiagnosticServiceInstances`) is done via the attribute `DiagnosticControlDTCSetting.dtcSettingParameter`. ]  
([RS\\_DEXT\\_00021](#), [RS\\_DEXT\\_00051](#))

<b>Class</b>	<b>DiagnosticControlDTCSetting</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ControlDTCSetting			
<b>Note</b>	This represents an instance of the "Control DTC Setting" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticControlDtcSettings			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dtcSettingClass	<a href="#">DiagnosticControlDTCSettingClass</a>	1	ref	This reference substantiates that abstract reference in the role <code>serviceClass</code> for this specific concrete class. Thereby, the the reference represents the ability to access shared attributes among all <code>DiagnosticControlDTCSetting</code> in the given context.
dtcSettingParameter	PositiveInteger	1	attr	This represents the <code>DTCSettingType</code> defined by ISO 14229-1. The pre-defined values are 1 (ON) and 2 (OFF).

Table 5.65: `DiagnosticControlDTCSetting`

<b>Class</b>	<b>DiagnosticControlDTCSettingClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ControlDTCSetting			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Control DTC Setting" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticControlDtcSettings			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
controlOptionRecordPresent	Boolean	1	attr	This represents the decision whether the DTCSettingControlOptionRecord (see ISO 14229-1) is in general supported in the request message.

**Table 5.66: DiagnosticControlDTCSettingClass**

### 5.5.10 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.

Each `DiagnosticResponseOnEvent` provides the ability to define a collection of triggers (modelled by means of the abstract meta-class `DiagnosticResponseOnEventTrigger`) that cause the sending of a response message.

The actual trigger behavior is defined by the sub-class of `DiagnosticResponseOnEventTrigger` used to specify whether the trigger shall be created in response to a data change or in response to a DTC change.

**[TPS\_DEXT\_01033] Semantics of triggers in the context of a `DiagnosticResponseOnEvent`** [ The semantics of a trigger in the context of a `DiagnosticResponseOnEvent` can be defined in two ways:

- The meta-class `DiagnosticDataChangeTrigger` allows for defining a trigger that activates on the change of the value of the referenced (in the role `dataIdentifier`) `DiagnosticDataIdentifier`.
- The meta-class `DiagnosticDtcChangeTrigger` allows for defining a trigger for the activation of the service. The entire proceedings of how the trigger activates and what DTCs are affected is managed at run-time and therefore no further configuration is required at this point.

] ([RS\\_DEXT\\_00022](#))

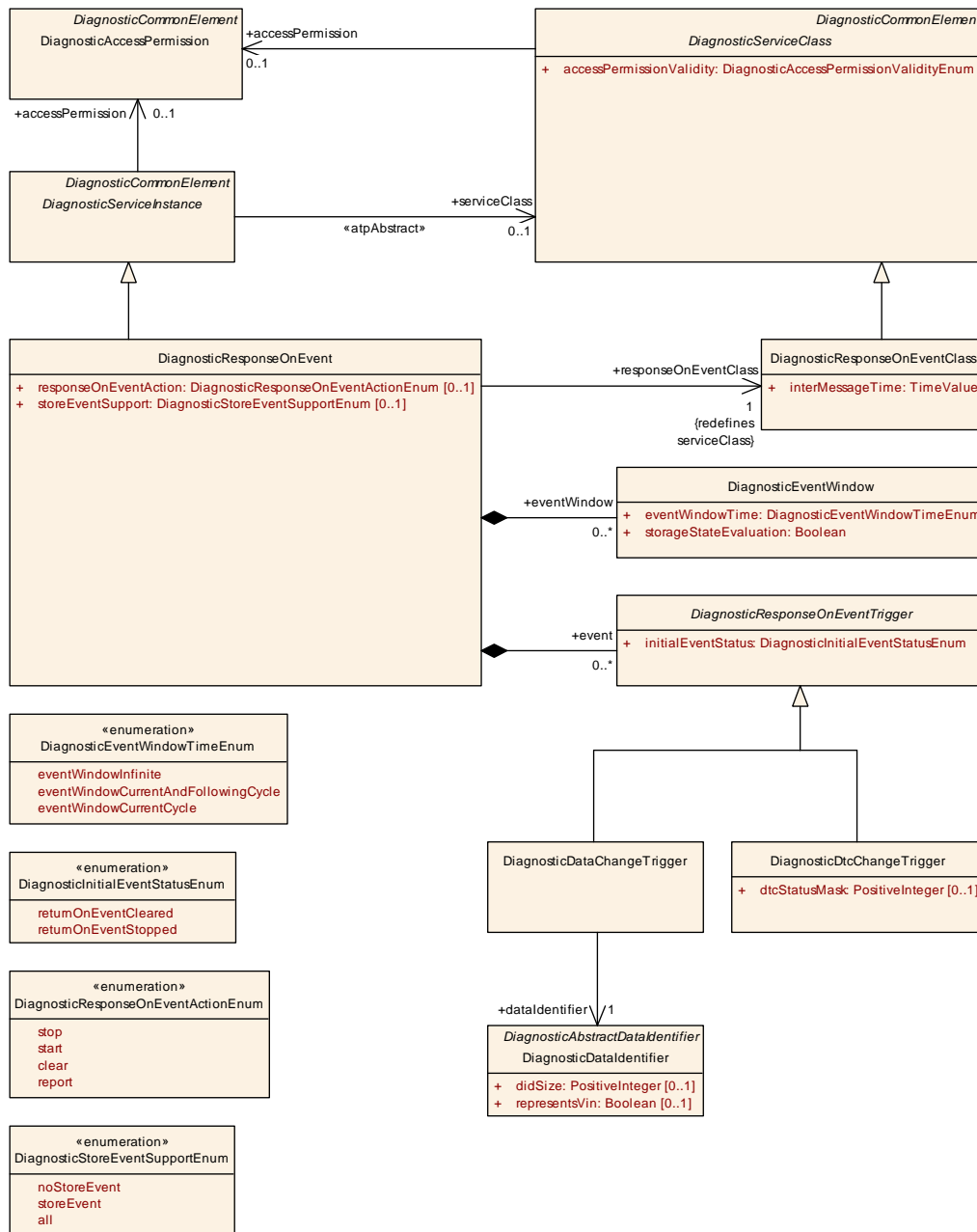


Figure 5.21: Modeling of diagnostic service ResponseOnEvent (0x86)

**[constr\_1365] Multiplicity of `DiagnosticResponseOnEvent.event`** [ The multiplicity of `DiagnosticResponseOnEvent.event` shall not exceed the upper bound 255. ]()

**[constr\_1366] Event ID in the context of diagnostic service ResponseOnEvent shall be unique** [ The value of `DiagnosticResponseOnEvent.event.dataIdentifier.id` shall be unique within the context of a given `DiagnosticResponseOnEvent`. ]()

**[constr\_1633] Existence of `DiagnosticResponseOnEvent.event` vs. `DiagnosticResponseOnEvent.responseOnEventAction`** [ The existence of attributes

[DiagnosticResponseOnEvent.event](#) vs. [DiagnosticResponseOnEvent.responseOnEventAction](#) is mutually exclusive, i.e. one shall only exist if the other does not exist.  $\perp()$

<b>Class</b>	<b>DiagnosticResponseOnEvent</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent			
<b>Note</b>	This represents an instance of the "Response on Event" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticResponseOnEvents			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
event	<a href="#">DiagnosticResponseOnEventTrigger</a>	*	aggr	This represents the collection of DiagnosticResponseOnEventTriggers defined in the context of the enclosing DiagnosticResponseOnEvent.
eventWindow	<a href="#">DiagnosticEventWindow</a>	*	aggr	This represents the applicable DiagnosticEventWindows
responseOnEventAction	<a href="#">DiagnosticResponseOnEventActionEnum</a>	0..1	attr	Defines sub-functions of the service ResponseOnEvent.
responseOnEventClass	<a href="#">DiagnosticResponseOnEventClass</a>	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.
storeEventSupport	<a href="#">DiagnosticStoreEventSupportEnum</a>	0..1	attr	Defines how a specific event shall be handled.

**Table 5.67: DiagnosticResponseOnEvent**

<b>Class</b>	<b>DiagnosticResponseOnEventClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent			
<b>Note</b>	This represents the ability to define common properties for alle instances of the "Response on Event" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticResponseOnEvents			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
interMessageTime	TimeValue	1	attr	Provide the minimum time in seconds between two consecutive transmissions of an ROE event.

**Table 5.68: DiagnosticResponseOnEventClass**

<b>Class</b>	<b>DiagnosticEventWindow</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent			
<b>Note</b>	This represents the ability to define the characteristics of the applicable event window			
<b>Base</b>	<a href="#">ARObject</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
eventWindowTime	<a href="#">DiagnosticEventWindowTimeEnum</a>	1	attr	This attribute clarifies the validity of the eventWindow
storageStateEvaluation	Boolean	1	attr	If this attribute is set to TRUE the StorageStateBit will be evaluated if this EventWindowTime is requested.

**Table 5.69: DiagnosticEventWindow**



<b>Class</b>	<i>DiagnosticResponseOnEventTrigger</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent			
<b>Note</b>	This represents the ability to further specify the events that are associated with the execution of the diagnostic service.			
<b>Base</b>	<i>ARObject</i>			
<b>Subclasses</b>	<a href="#">DiagnosticDataChangeTrigger</a> , <a href="#">DiagnosticDtcChangeTrigger</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
initialEvent Status	<a href="#">DiagnosticInitialEventStatusEnum</a>	1	attr	This represents the initial status of the enclosing DiagnosticResponseOnEventTrigger.

**Table 5.70: DiagnosticResponseOnEventTrigger**

<b>Class</b>	<b>DiagnosticDataChangeTrigger</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent			
<b>Note</b>	This represents the ability to define a trigger based on the change of a given DiagnosticDataIdentifier.			
<b>Base</b>	<i>ARObject</i> , <a href="#">DiagnosticResponseOnEventTrigger</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataIdentifier	<a href="#">DiagnosticDataIdentifier</a>	1	ref	This represents the corresponding DiagnosticData Identifier.

**Table 5.71: DiagnosticDataChangeTrigger**

<b>Class</b>	<b>DiagnosticDtcChangeTrigger</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent			
<b>Note</b>	This represents the ability to define a trigger that executes on the change of any DiagnosticTroubleCode.			
<b>Base</b>	<i>ARObject</i> , <a href="#">DiagnosticResponseOnEventTrigger</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dtcStatusMask	PositiveInteger	0..1	attr	This attribute represents the ability to define a status mask for the triggering of an ROE response on the change of a DTC.

**Table 5.72: DiagnosticDtcChangeTrigger**

<b>Enumeration</b>	<b>DiagnosticInitialEventStatusEnum</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent			
<b>Note</b>	This represents the ability to define an initial status for the ROE service.			
<b>Literal</b>	<b>Description</b>			
returnOnEvent Cleared	This means that the ResponseOnEvent is initially cleared. <b>Tags:</b> atp.EnumerationValue=0			
returnOnEvent Stopped	This means that the ResponseOnEvent is initially stopped. <b>Tags:</b> atp.EnumerationValue=1			

**Table 5.73: DiagnosticInitialEventStatusEnum**

<b>Enumeration</b>	<b>DiagnosticEventWindowTimeEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent
<b>Note</b>	This represents the ability to define the semantics of the event window.
<b>Literal</b>	<b>Description</b>
eventWindow CurrentAnd FollowingCycle	This means that the window extends to this and the following cycle. <b>Tags:</b> atp.EnumerationValue=0
eventWindow CurrentCycle	This means that the window is limited to the current cycle. <b>Tags:</b> atp.EnumerationValue=1
eventWindowInfinite	This means that the window extends without a border. <b>Tags:</b> atp.EnumerationValue=2

**Table 5.74: DiagnosticEventWindowTimeEnum**

<b>Enumeration</b>	<b>DiagnosticStoreEventSupportEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent
<b>Note</b>	This meta-class has the ability to define how a specific event in the context of the UDS service ResponseOnEvent shall be handled.
<b>Literal</b>	<b>Description</b>
all	The server supports both, storing and not storing the event. <b>Tags:</b> atp.EnumerationValue=2
noStoreEvent	The event terminates when the server is powered down. <b>Tags:</b> atp.EnumerationValue=0
storeEvent	The event is persisted over a power down cycle. <b>Tags:</b> atp.EnumerationValue=1

**Table 5.75: DiagnosticStoreEventSupportEnum**

<b>Enumeration</b>	<b>DiagnosticResponseOnEventActionEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent
<b>Note</b>	This meta-class has the ability to define sub-functions of the UDS service ResponseOnEvent.
<b>Literal</b>	<b>Description</b>
clear	Clears the configured events. <b>Tags:</b> atp.EnumerationValue=2
report	Reports the activated events. <b>Tags:</b> atp.EnumerationValue=3
start	Starts the response on event service. <b>Tags:</b> atp.EnumerationValue=1
stop	Stops the response on event service. <b>Tags:</b> atp.EnumerationValue=0

**Table 5.76: DiagnosticResponseOnEventActionEnum**

### 5.5.11 ReadDTCInformation

This chapter describes the modeling of diagnostic services `ReadDTCInformation` (0x19). The purpose of this service is enable a tester to read a Diagnostic Trouble Code from the AUTOSAR Dcm [10] (that, in turn, fetches the information from the AUTOSAR Dem [11]).

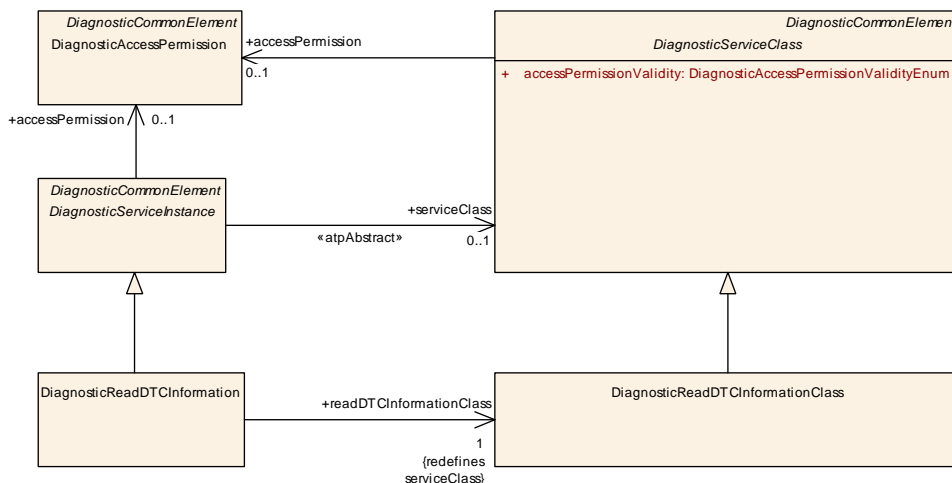


Figure 5.22: Modeling of diagnostic service `ReadDTCInformation` (0x19)

**[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].** ]([RS\\_DEXT\\_00006](#), [RS\\_DEXT\\_00051](#))

**[TPS\_DEXT\_01060] Applicable values for `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\_MIRROR\_MEMORY\_DTC\_BY\_STATUS\_MASK
- REPORT\_NUMBER\_OF\_MIRROR\_MEMORY\_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\_MIRROR\_MEMORY\_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\_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

The meanings of these values are described in the applicable ISO document (ISO 14229-1) [16]. ]([RS\\_DEXT\\_00001](#), [RS\\_DEXT\\_00006](#), [RS\\_DEXT\\_00051](#))

Please note that there is nothing to configure for [DiagnosticReadDTCInformation](#) beyond its mere existence.

<b>Class</b>	<b>DiagnosticReadDTCInformation</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ReadDTCInformation			
<b>Note</b>	This represents an instance of the "Read DTC Information" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticReadDtcInformations			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>





Class	DiagnosticReadDTCInformation			
read DTCInformation Class	<a href="#">DiagnosticReadDTCInformationClass</a>	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.

**Table 5.77: DiagnosticReadDTCInformation**

Class	DiagnosticReadDTCInformationClass			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ReadDTCInformation			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "ReadDTCInformation" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticReadDtcInformations			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
-	-	-	-	-

**Table 5.78: DiagnosticReadDTCInformationClass**

### 5.5.12 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 `DiagnosticRoutine`, on the diagnostic stack at the request of the tester.

`DiagnosticRoutines` consist of up to three possible components:

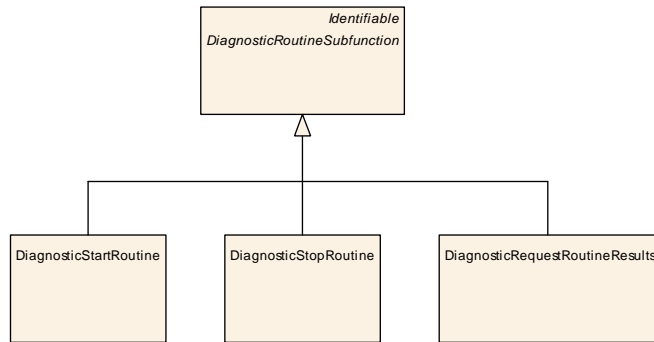
- Start Routine
- Stop Routine
- Request Routine Results

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](#)

]([RS\\_DEXT\\_00015](#))



**Figure 5.23: Abstract base class for subfunctions of diagnostic service RoutineControl (0x31)**

<b>Class</b>	<i>DiagnosticRoutineSubfunction</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
<b>Note</b>	This meta-class acts as an abstract base class to routine subfunctions.			
<b>Base</b>	ARObject, <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">DiagnosticRequestRoutineResults</a> , <a href="#">DiagnosticStartRoutine</a> , <a href="#">DiagnosticStopRoutine</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
access Permission	<a href="#">DiagnosticAccessPermission</a>	0..1	ref	This reference represents the access permission of the owning routine subfunction.

**Table 5.79: DiagnosticRoutineSubfunction**

[TPS\_DEXT\_01088] **Semantics of [DiagnosticRoutine.id](#)** [ The attribute [DiagnosticRoutine.id](#) represents the so-called *identifier* of the [DiagnosticRoutine](#). ]([RS\\_DEXT\\_00036](#))

[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. ]([RS\\_DEXT\\_00015](#), [RS\\_DEXT\\_00051](#))

The sub-functions actually have different properties i.e. the arguments to a [DiagnosticRoutine](#)) that require a dedicated modeling for this purpose.

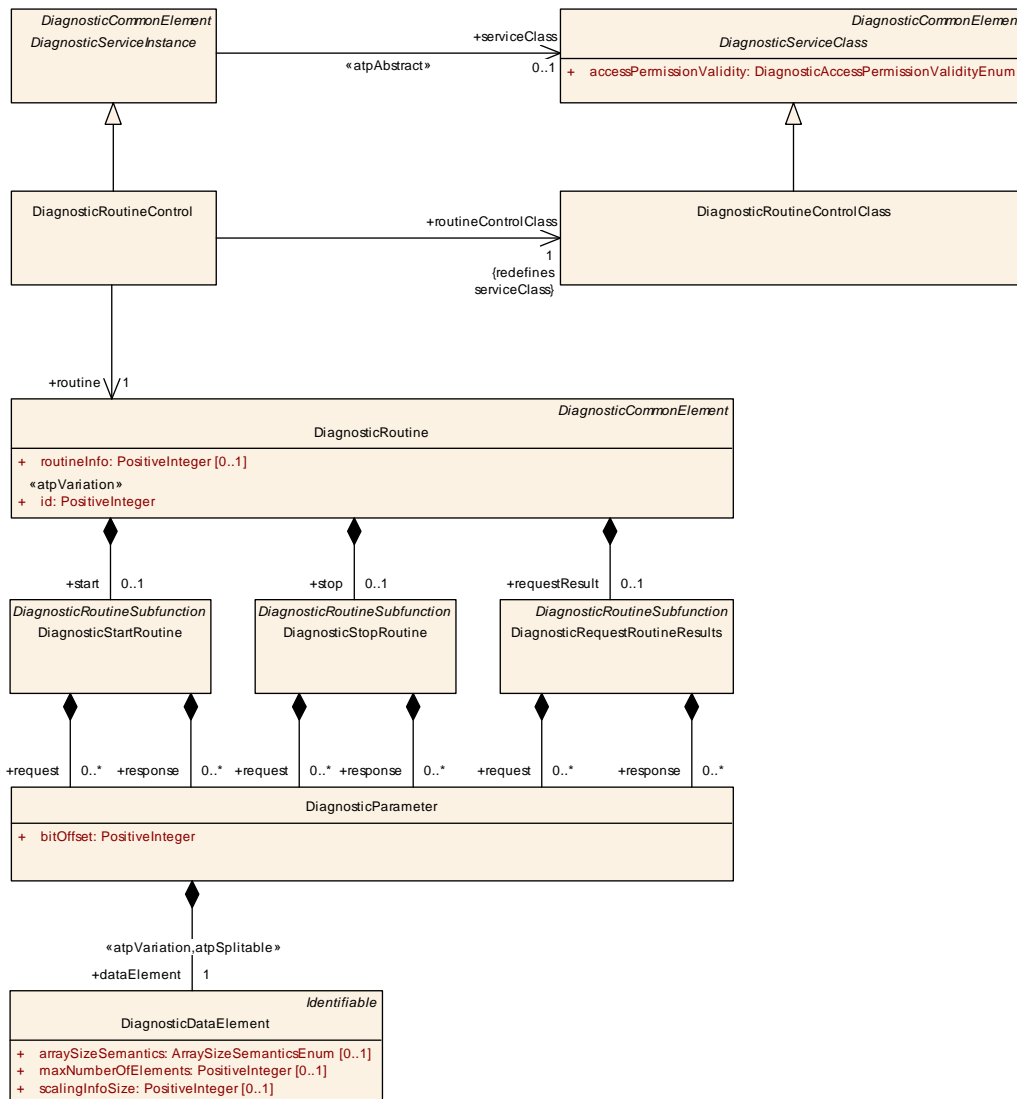


Figure 5.24: Modeling of diagnostic service RoutineControl (0x31)

[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.response**
- **DiagnosticRequestRoutineResults.request**
- **DiagnosticRequestRoutineResults.response**

](RS\_DEXT\_00015)

A `DiagnosticParameter`, in turn, aggregates a `DiagnosticDataElement` (see section 4.2) in the role `dataElement`.

**[TPS\_DEXT\_01080] Diagnostic Routine needs to be started** [ ISO 14229-1 [16] 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. ](*RS\_DEXT\_00015*)

**[constr\_1339] Existence of `DiagnosticRoutine.start`** [ In a complete `DiagnosticExtract`, the attribute `DiagnosticRoutine.start` shall always exist for any given `DiagnosticRoutine`. ]()

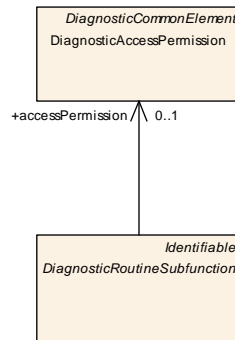
**[TPS\_DEXT\_01035] Existence of `DiagnosticRoutine.stop` and `DiagnosticRoutine.requestResult`** [ In contrast to `DiagnosticRoutine.start` (as clarified by [constr\_1339]), the existence of `DiagnosticRoutine.stop` and `DiagnosticRoutine.requestResult` is truly optional. ](*RS\_DEXT\_00015*)

**[constr\_1340] Consistency of `DiagnosticServiceSwMapping` with respect to synchronously called `DiagnosticRoutines`** [ 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`** [ 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`. ]()

**[TPS\_DEXT\_01049] Consistency of `DiagnosticServiceSwMapping` with respect to routine IDs** [ For each `DiagnosticServiceSwMapping` that references a `DiagnosticRoutineNeeds` and a `DiagnosticRoutineControl`, the value of `DiagnosticRoutineNeeds.ridNumber` shall be ignored and the value of `DiagnosticRoutineControl.routine.id` shall be taken instead. ](*RS\_DEXT\_00015*, *RS\_DEXT\_00052*)





**Figure 5.25: Permission is granted per subfunction**

Access permissions for service `RoutineControl` are granted per subfunction, as opposed to the entire service (i.e. `DiagnosticRoutineControl`).

For this purpose, the reference from the abstract base class<sup>1</sup> `DiagnosticRoutineSubfunction` to `DiagnosticAccessPermission` in the role `accessPermission` exists (see Figure 5.25).

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

**[constr\_1612] Reference from `DiagnosticRoutineControl` to `DiagnosticAccessPermission` has no meaning** [ The reference from `DiagnosticRoutineControl` (via its abstract base class `DiagnosticServiceInstance`) in the role `accessPermission` to meta-class `DiagnosticAccessPermission` shall not be used. ]()

<b>Class</b>	<b>DiagnosticRoutine</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
<b>Note</b>	This meta-class represents the ability to define a diagnostic routine. <b>Tags:</b> atp.recommendedPackage=DiagnosticRoutines			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
id	PositiveInteger	1	attr	This is the numerical identifier used to identify the <code>DiagnosticRoutine</code> in the scope of diagnostic workflow <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime
requestResult	<a href="#">DiagnosticRequestRoutineResults</a>	0..1	aggr	This represents the ability to request the result of a running routine.



<sup>1</sup>of meta-classes `DiagnosticStartRoutine`, `DiagnosticStopRoutine`, and `DiagnosticRequestRoutineResults`



Class		DiagnosticRoutine		
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	<a href="#">DiagnosticStartRoutine</a>	0..1	aggr	This represents the ability to start a routine
stop	<a href="#">DiagnosticStopRoutine</a>	0..1	aggr	This represents the ability to stop a running routine.

**Table 5.80: DiagnosticRoutine**

Class		DiagnosticStartRoutine		
Package		M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics		
Note		This represents the ability to start a diagnostic routine.		
Base		<i>ARObject</i> , <a href="#">DiagnosticRoutineSubfunction</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>		
Attribute	Type	Mul.	Kind	Note
request	<a href="#">DiagnosticParameter</a>	*	aggr	This represents the request parameters.
response	<a href="#">DiagnosticParameter</a>	*	aggr	This represents the response parameters.

**Table 5.81: DiagnosticStartRoutine**

Class		DiagnosticStopRoutine		
Package		M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics		
Note		This represents the ability to stop a diagnostic routine.		
Base		<i>ARObject</i> , <a href="#">DiagnosticRoutineSubfunction</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>		
Attribute	Type	Mul.	Kind	Note
request	<a href="#">DiagnosticParameter</a>	*	aggr	This represents the request parameters.
response	<a href="#">DiagnosticParameter</a>	*	aggr	This represents the response parameters.

**Table 5.82: DiagnosticStopRoutine**

Class		DiagnosticRequestRoutineResults		
Package		M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics		
Note		This meta-class represents the ability to define the result of a diagnostic routine execution.		
Base		<i>ARObject</i> , <a href="#">DiagnosticRoutineSubfunction</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>		
Attribute	Type	Mul.	Kind	Note
request	<a href="#">DiagnosticParameter</a>	*	aggr	This represents the request parameters.
response	<a href="#">DiagnosticParameter</a>	*	aggr	This represents the response parameters.

**Table 5.83: DiagnosticRequestRoutineResults**

<b>Class</b>	<b>DiagnosticRoutineControl</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::RoutineControl			
<b>Note</b>	This represents an instance of the "Routine Control" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticRoutineControls			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
routine	<a href="#">DiagnosticRoutine</a>	1	ref	This refers to the applicable DiagnosticRoutine.
routineControl Class	<a href="#">DiagnosticRoutineControlClass</a>	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.

**Table 5.84: DiagnosticRoutineControl**

<b>Class</b>	<b>DiagnosticRoutineControlClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::RoutineControl			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Routine Control" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticRoutineControls			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 5.85: DiagnosticRoutineControlClass**

<b>Class</b>	<b>DiagnosticRoutineNeeds</b>			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
<b>Note</b>	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.			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">DiagnosticCapabilityElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a> , <a href="#">ServiceNeeds</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
diagRoutine Type	<a href="#">DiagnosticRoutineTypeEnum</a>	1	attr	This denotes the type of diagnostic routine which is implemented by the referenced server port.
ridNumber	PositiveInteger	0..1	attr	This represents a routine identifier for the diagnostic routine.  This allows to predefine the RID number if the a function developer has received a particular requirement from the OEM or from a standardization body.

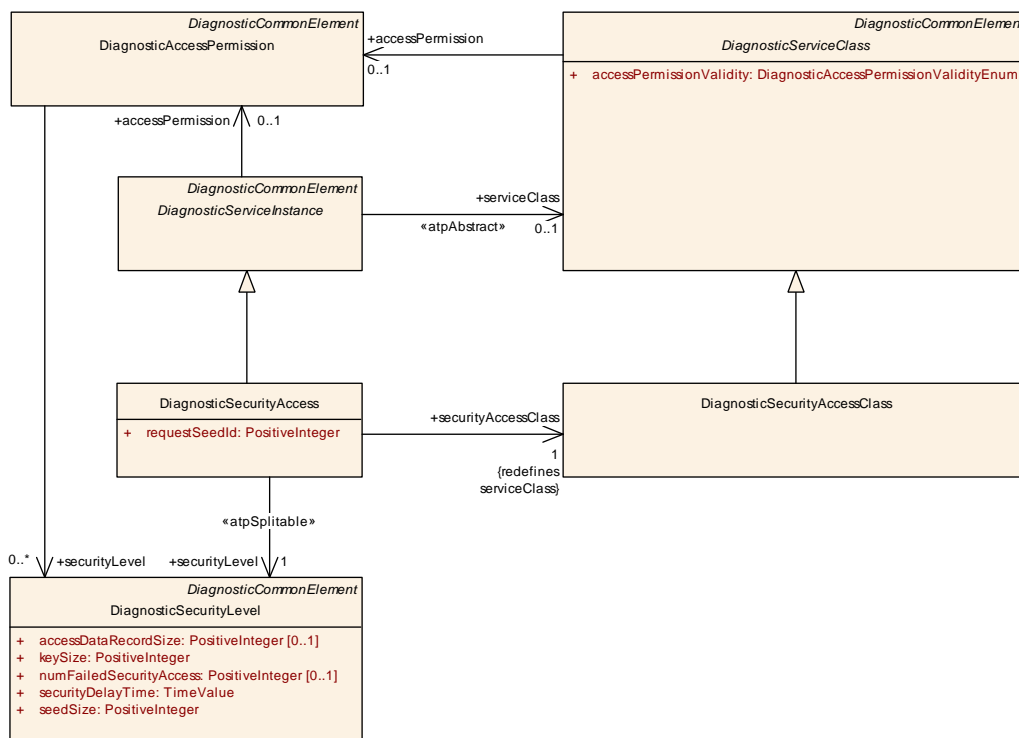
**Table 5.86: DiagnosticRoutineNeeds**

<b>Enumeration</b>	<b>DiagnosticRoutineTypeEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds
<b>Note</b>	This enumerator specifies the different types of diagnostic routines.
<b>Literal</b>	<b>Description</b>
asynchronous	This indicates that the diagnostic server is not blocked while the diagnostic routine is running. <b>Tags:</b> atp.EnumerationValue=0
synchronous	This indicates that the diagnostic routine blocks the diagnostic server in the ECU while the routine is running. <b>Tags:</b> atp.EnumerationValue=1

**Table 5.87: DiagnosticRoutineTypeEnum**

### 5.5.13 SecurityAccess

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



**Figure 5.26: Modeling of diagnostic service 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. ] ([RS\\_DEXT\\_00009](#), [RS\\_DEXT\\_00042](#))

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 where ECU configuration is generated from the `DiagnosticExtract` the reference is needed to be able to make sense of the model for the given `DiagnosticSecurityAccess`.

Please note that (as already explained in section 5.5) 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 ISO 14229-1 [16] 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 ISO 14229-1 [16]. ]([RS\\_DEXT\\_00009](#))

However, there is further dimension to take into account, namely the `DiagnosticSecurityLevel`. According to ISO 14229-1 [16], 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. ]([RS\\_DEXT\\_00009](#))

**[constr\_1342] Possible values for `DiagnosticSecurityAccess.requestSeedId`** [ The value of the attribute `DiagnosticSecurityAccess.requestSeedId` shall only be set to an odd number<sup>2</sup>.

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 ISO 26021-2 [17]*)
- all odd numbers in the closed interval **0x61 .. 0x7E**

]()

In contrast to a similar situation in the case of the diagnostic service `SessionControl` (see section 5.5.14), 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`.

---

<sup>2</sup>The even numbers are reserved for the identification of the corresponding *sendKey* sub-function, as explained by [\[TPS\\_DEXT\\_01036\]](#).

[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](#). ]  
([RS\\_DEXT\\_00002](#), [RS\\_DEXT\\_00009](#), [RS\\_DEXT\\_00042](#))

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

<b>Class</b>	<b>DiagnosticSecurityAccess</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::SecurityAccess			
<b>Note</b>	This represents an instance of the "Security Access" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticSecurityAccesses			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
requestSeedId	PositiveInteger	1	attr	This would be 0x01, 0x03, 0x05, ... The sendKey id can be computed by adding 1 to the requestSeedId
securityAccessClass	<a href="#">DiagnosticSecurityAccessClass</a>	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 <a href="#">DiagnosticSecurityAccess</a> in the given context.
securityLevel	<a href="#">DiagnosticSecurityLevel</a>	1	ref	This reference identifies the applicable security level for the security access. <b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=securityLevel

**Table 5.88: DiagnosticSecurityAccess**

<b>Class</b>	<b>DiagnosticSecurityAccessClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::SecurityAccess			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Security Access" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticSecurityAccesses			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 5.89: DiagnosticSecurityAccessClass**

## 5.5.14 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.

[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**. ] (RS\_DEXT\_00003, RS\_DEXT\_00040)

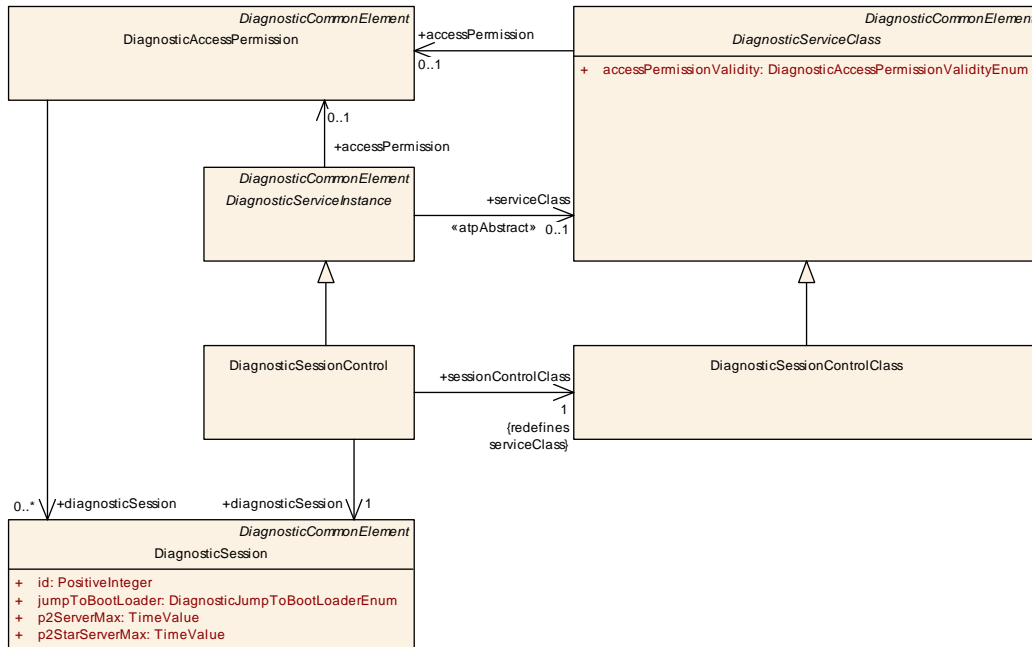


Figure 5.27: Modeling of diagnostic service **SessionControl** (0x10)

According to ISO 14229-1 [16], 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**. ] (RS\_DEXT\_00003, RS\_DEXT\_00051)

[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<sup>3</sup> for diagnostic communication. ]  
([RS\\_DEXT\\_00003](#), [RS\\_DEXT\\_00040](#))

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`.

<b>Class</b>	<b>DiagnosticSessionControl</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::SessionControl			
<b>Note</b>	This represents an instance of the "Session Control" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticSessionControls			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
diagnostic Session	<a href="#">DiagnosticSession</a>	1	ref	This represents the applicable DiagnosticSessions
sessionControl Class	<a href="#">DiagnosticSessionControlClass</a>	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.

**Table 5.90: DiagnosticSessionControl**

<b>Class</b>	<b>DiagnosticSessionControlClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::SessionControl			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Session Control" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticSessionControls			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
—	—	—	—	—

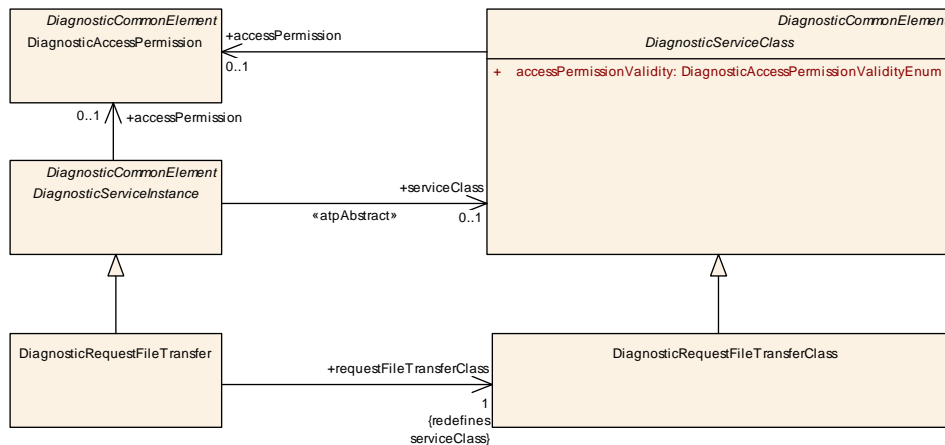
**Table 5.91: DiagnosticSessionControlClass**

### 5.5.15 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.

<sup>3</sup>This is similar to the definition of commonly used data types in a software development project





**Figure 5.28: Modeling of diagnostic service RequestFileTransfer (0x38)**

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`. ] ([RS\\_DEXT\\_00057](#))

<b>Class</b>	<b>DiagnosticRequestFileTransfer</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::RequestFileTransfer			
<b>Note</b>	This diagnostic service instance implements the UDS service 0x38. <b>Tags:</b> atp.recommendedPackage=DiagnosticRequestFileTransfers			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
requestFileTransferClass	<a href="#">DiagnosticRequestFileTransferClass</a>	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 <code>DiagnosticRequestFileTransfer</code> in the given context.

**Table 5.92: DiagnosticRequestFileTransfer**

<b>Class</b>	<b>DiagnosticRequestFileTransferClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::RequestFileTransfer			
<b>Note</b>	This meta-class contains attributes shared by all instances of the "Request File transfer" diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticRequestFileTransfers			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 5.93: DiagnosticRequestFileTransferClass**

## 5.6 OBD Diagnostic Services supported by AUTOSAR

Support for diagnostic services for on-board diagnostics (OBD) [18] requires the definition of further meta-classes similar to e.g. `DiagnosticDataIdentifier` 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 `DiagnosticParameterIdentifier` defines the following properties:

- Each `DiagnosticParameterIdentifier` gets a numerical ID carried in the attribute `DiagnosticParameterIdentifier.id`.
- `DiagnosticParameterIdentifier` may also contains 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`.

](*RS\_DEXT\_00068*)

As already mentioned in [TPS\_DEXT\_01092], the `DiagnosticParameterIdentifier` 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 `DiagnosticParameterIdentifier`.

**[constr\_1447] Restrictions for the value of `DiagnosticParameterIdentifier.id`** [ The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of `DiagnosticParameterIdentifier.id`. ]()

**[constr\_1448] Interval of `DiagnosticParameterIdentifier.id`** [ The allowed interval for values of `DiagnosticParameterIdentifier.id` shall not exceed [0..255]. ]()

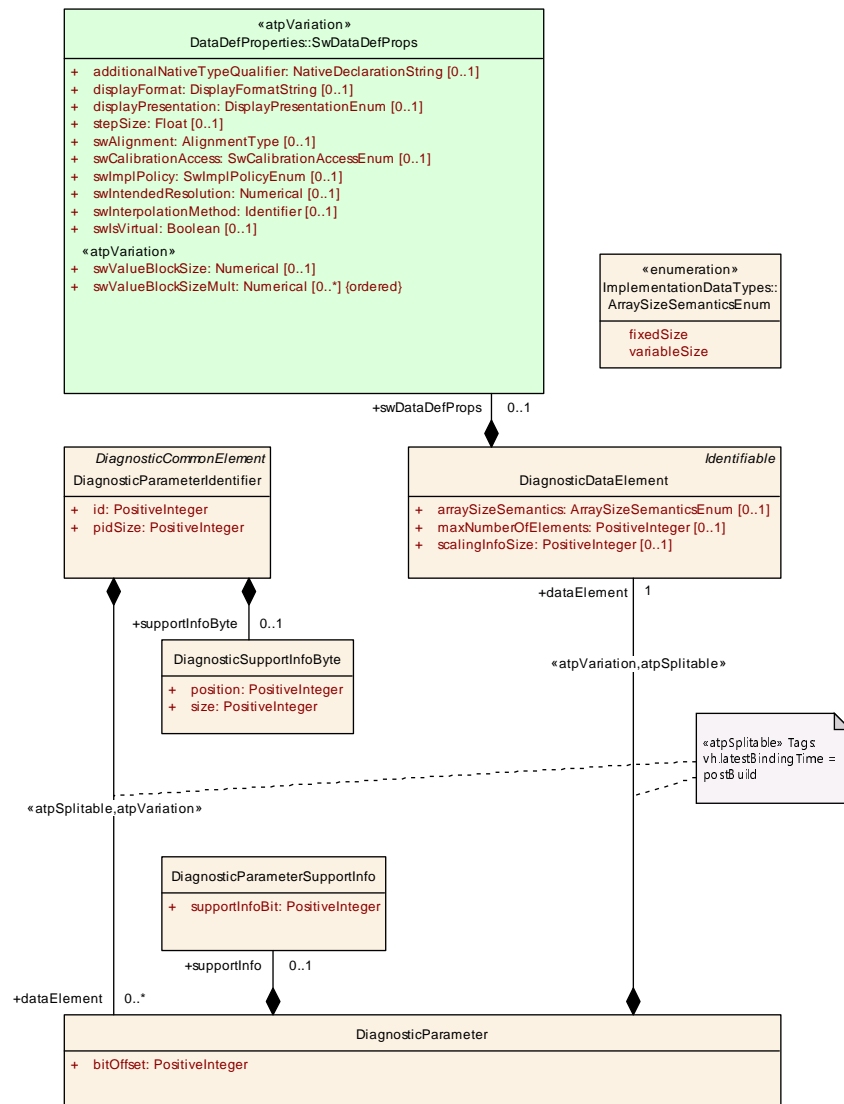


Figure 5.29: Modeling of the **DiagnosticParameterIdentifier**

[constr\_1449] PID shall only carry a fixed-length collection of data [ The value of `DiagnosticParameterIdentifier.dataElement.dataElement.arraySizeSemantics` shall not be set to `variableSize`. ]()



<b>Class</b>	<b>DiagnosticParameterIdentifier</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
<b>Note</b>	<p>This meta-class represents the ability to model a diagnostic parameter identifier (PID) for the purpose of executing on-board diagnostics (OBD).</p> <p><b>Tags:</b> atp.recommendedPackage=DiagnosticParameterIdentifiers</p>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataElement	<a href="#">DiagnosticParameter</a>	*	aggr	<p>This represents the data carried by the Diagnostic ParameterIdentifier.</p> <p><b>Stereotypes:</b> atpSplittable; atpVariation</p> <p><b>Tags:</b> atp.Splitkey=dataElement, variationPoint.shortLabel</p> <p>vh.latestBindingTime=postBuild</p>





Class		DiagnosticParameterIdentifier		
id	PositiveInteger	1	attr	This is the numerical identifier used to identify the DiagnosticParameterIdentifier in the scope of diagnostic workflow (see SAE J1979-DA).
pidSize	PositiveInteger	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.
supportInfoByte	<a href="#">DiagnosticSupportInfo Byte</a>	0..1	aggr	This represents the supported information associated with the DiagnosticParameterIdentifier.

**Table 5.94: DiagnosticParameterIdentifier**

Class		DiagnosticParameterSupportInfo		
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This represents a way to define which bit of the supportInfo is representing this part of the PID			
Base	ARObject			
Attribute	Type	Mul.	Kind	Note
supportInfoBit	PositiveInteger	1	attr	defines the bit in the SupportInfo byte, which represents the PID DataElement pidSize / position / size. Unit: byte.

**Table 5.95: DiagnosticParameterSupportInfo**

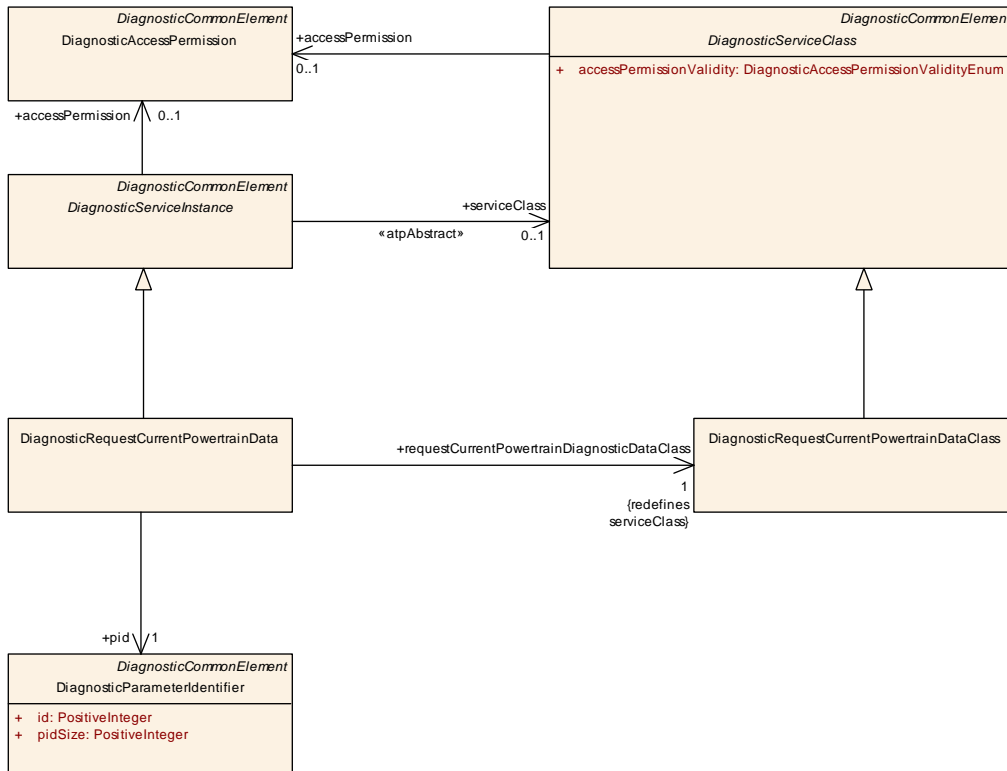
Class		DiagnosticSupportInfoByte		
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class defines the support information (typically byte A) to declare the usability of the Data Elements within the so-called packeted PIDs (e.g. PID\$68).			
Base	ARObject			
Attribute	Type	Mul.	Kind	Note
position	PositiveInteger	1	attr	This represents the position of the supportInfo in the PID. Unit: byte.
size	PositiveInteger	1	attr	This represents the size of the supportInfo within the PID. Unit: byte.

**Table 5.96: DiagnosticSupportInfoByte**

### 5.6.1 OBD Mode 0x01 (RequestCurrentPowertrainDiagnosticData)

The service RequestCurrentPowertrainDiagnosticData is modeled in Figure 5.30.

**[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. ] ([RS\\_DEXT\\_00069](#))



**Figure 5.30: Modeling of diagnostic service for OBD Mode 0x01**

<b>Class</b>	<b>DiagnosticRequestCurrentPowertrainData</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x01_RequestCurrentPowertrainDiagnosticData			
<b>Note</b>	This meta-class represents the ability to model an instance of the OBD mode 0x01 service. <b>Tags:</b> atp.recommendedPackage=DiagnosticRequestCurrentPowertrainData			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
pid	<a href="#">DiagnosticParameterIdentifier</a>	1	ref	This represents the PID associated with this instance of the OBD mode 0x01 service.
requestCurrentPowertrainDiagnosticDataClass	<a href="#">DiagnosticRequestCurrentPowertrainDataClass</a>	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 DiagnosticRequestCurrentPowertrainData in the given context.

**Table 5.97: DiagnosticRequestCurrentPowertrainData**

<b>Class</b>	<b>DiagnosticRequestCurrentPowertrainDataClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x01_RequestCurrentPowertrainDiagnosticData			





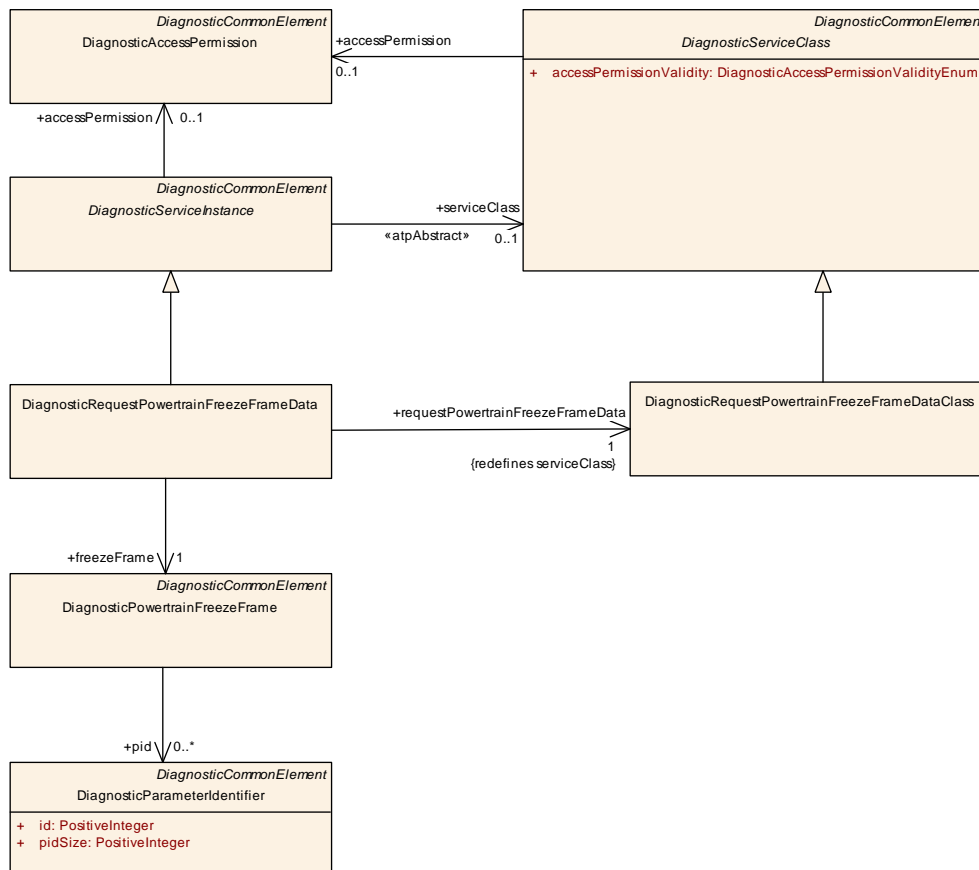
<b>Class</b>	<b>DiagnosticRequestCurrentPowertrainDataClass</b>			
<b>Note</b>	This meta-class represents the ability to define common properties for all instances of the "Request current Powertrain Data" OBD diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticRequestCurrentPowertrainData			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
-	-	-	-	-

**Table 5.98: DiagnosticRequestCurrentPowertrainDataClass**

### 5.6.2 OBD Mode 0x02 (RequestPowertrainFreezeFrameData)

The service `RequestPowertrainFreezeFrameData` is modeled in Figure 5.31.

**[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`. | ([RS\\_DEXT\\_00070](#))



**Figure 5.31: Modeling of diagnostic service for OBD Mode 0x02**

<b>Class</b>	<b>DiagnosticRequestPowertrainFreezeFrameData</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x02_RequestPowertrainFreezeFrameData			
<b>Note</b>	This meta-class represents the ability to model an instance of the OBD mode 0x02 service. <b>Tags:</b> atp.recommendedPackage=DiagnosticPowertrainFreezeFrames			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
freezeFrame	<a href="#">DiagnosticPowertrainFreezeFrame</a>	1	ref	This represents the associated freeze-frame.
requestPowertrainFreezeFrameData	<a href="#">DiagnosticRequestPowertrainFreezeFrameDataClass</a>	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.

**Table 5.99: DiagnosticRequestPowertrainFreezeFrameData**

<b>Class</b>	<b>DiagnosticRequestPowertrainFreezeFrameDataClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x02_RequestPowertrainFreezeFrameData			
<b>Note</b>	This meta-class represents the ability to define common properties for all instances of the "Request Powertrain Freeze Frame Data" OBD diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticPowertrainFreezeFrames			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 5.100: DiagnosticRequestPowertrainFreezeFrameDataClass**

<b>Class</b>	<b>DiagnosticPowertrainFreezeFrame</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x02_RequestPowertrainFreezeFrameData			
<b>Note</b>	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. <b>Tags:</b> atp.recommendedPackage=DiagnosticPowertrainFreezeFrames			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
pid	<a href="#">DiagnosticParameterIdentifier</a>	*	ref	This represents the PID associated with this instance of the OBD mode 0x02 service.

**Table 5.101: DiagnosticPowertrainFreezeFrame**



### 5.6.3 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**. ] (RS\_DEXT\_00071)

The modeling is sketched in Figure 5.32.

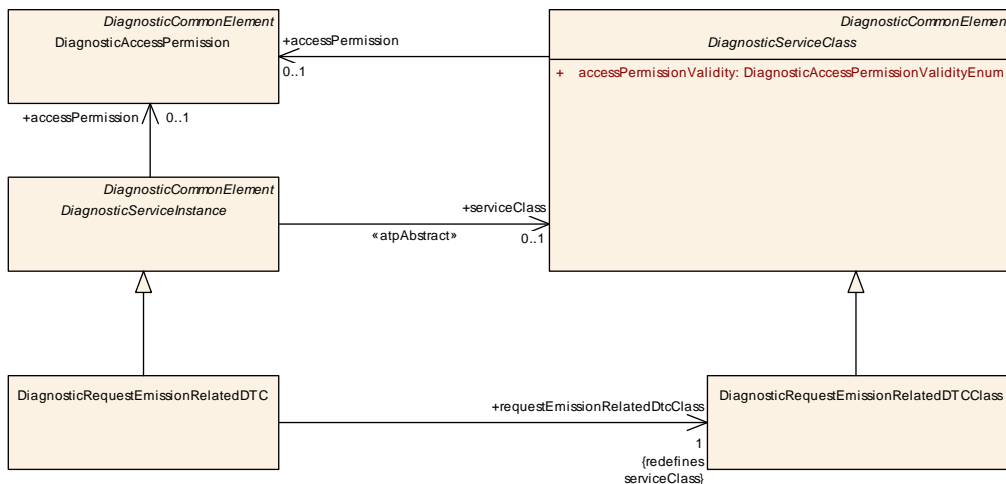


Figure 5.32: Modeling of diagnostic service for OBD Modes 0x03, 0x07

<b>Class</b>	<b>DiagnosticRequestEmissionRelatedDTC</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x03_0x07_RequestEmissionRelatedDTC			
<b>Note</b>	This meta-class represents the ability to model an instance of the OBD mode 0x03/0x07 service. <b>Tags:</b> atp.recommendedPackage=DiagnosticRequestEmissionRelatedDTCs			
<b>Base</b>	<i>ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
requestEmissionRelatedDtcClass	<a href="#">DiagnosticRequestEmissionRelatedDTCClass</a>	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 DiagnosticRequestEmissionRelatedDTC in the given context.

Table 5.102: DiagnosticRequestEmissionRelatedDTC

<b>Class</b>	<b>DiagnosticRequestEmissionRelatedDTCClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x03_0x07_RequestEmissionRelatedDTC			
<b>Note</b>	This meta-class represents the ability to define common properties for all instances of the "Request Emission Related DTC" OBD diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticRequestEmissionRelatedDTCs			





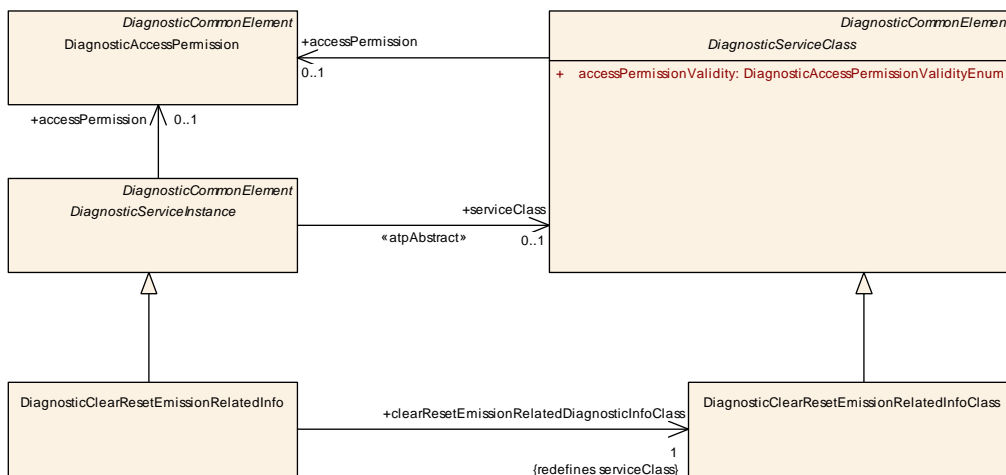
<b>Class</b>	<b>DiagnosticRequestEmissionRelatedDTCCClass</b>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
—	—	—	—	—

**Table 5.103: DiagnosticRequestEmissionRelatedDTCCClass**

### 5.6.4 OBD Mode 0x04 (ClearResetEmissionRelatedDiagnosticInformation)

[TPS\_DEXT\_01128] Semantics of meta-class [DiagnosticClearResetEmissionRelatedInfo](#) [ The OBD diagnostic service `ClearResetEmissionRelatedDiagnosticInformation` is supported by means of the meta-class [DiagnosticClearResetEmissionRelatedInfo](#). ] ([RS\\_DEXT\\_00072](#))

The modeling is sketched in Figure 5.33.



**Figure 5.33: Modeling of diagnostic service for OBD Mode 0x04**

<b>Class</b>	<b>DiagnosticClearResetEmissionRelatedInfo</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x04_ClearResetEmissionRelatedInfo			
<b>Note</b>	This meta-class represents the ability to model an instance of the OBD mode 0x04 service. <b>Tags:</b> atp.recommendedPackage=DiagnosticClearResetEmissionRelatedInfos			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>





Class	DiagnosticClearResetEmissionRelatedInfo			
clearResetEmissionRelatedDiagnosticInfoClass	<a href="#">DiagnosticClearResetEmissionRelatedInfoClass</a>	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 DiagnosticClearResetEmissionRelatedInfo in the given context.

**Table 5.104: DiagnosticClearResetEmissionRelatedInfo**

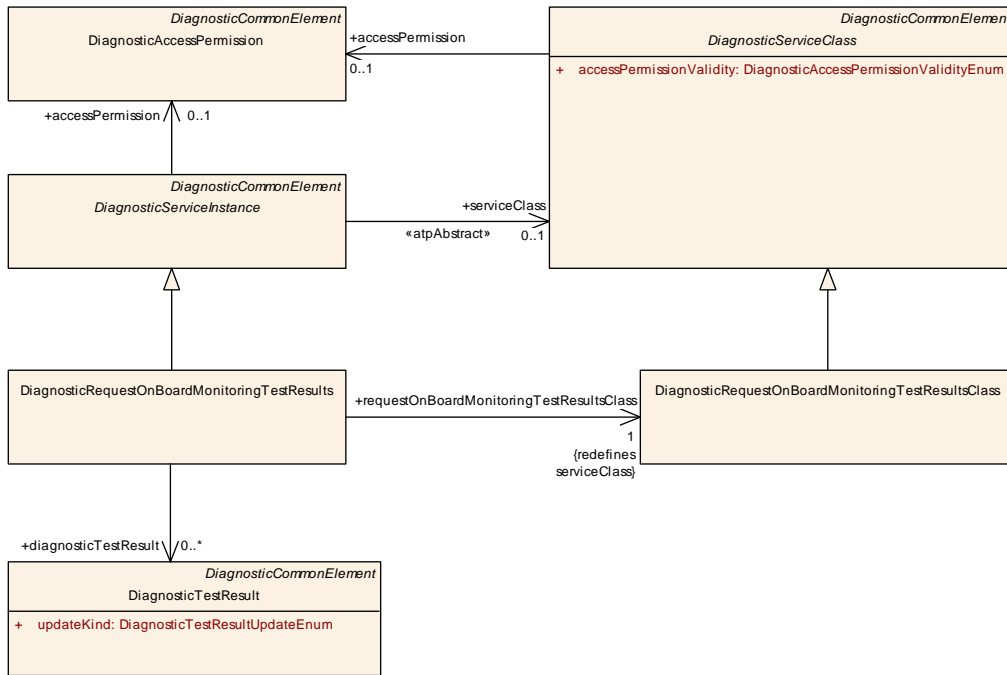
Class	DiagnosticClearResetEmissionRelatedInfoClass			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x04_ClearResetEmissionRelatedInfo			
Note	This meta-class represents the ability to define common properties for all instances of the "Clear Reset Emission Related Data" OBD diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticClearResetEmissionRelatedInfos			
Base	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
Attribute	Type	Mul.	Kind	Note
-	-	-	-	-

**Table 5.105: DiagnosticClearResetEmissionRelatedInfoClass**

### 5.6.5 OBD Mode 0x06 (RequestOnBoardMonitoringTestResults)

**[TPS\_DEXT\_01129] Support for OBD diagnostic service RequestOnBoardMonitoringTestResults** [ The OBD diagnostic service RequestOnBoardMonitoringTestResults is supported by means of meta-class [DiagnosticRequestOnBoardMonitoringTestResults](#) that refers to the representation of the test result modeled as meta-class [DiagnosticTestResult](#) ] ([RS\\_DEXT\\_00073](#))

The modeling is sketched in Figure 5.34.



**Figure 5.34: Modeling of diagnostic service for OBD Mode 0x06**

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** [ The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of DiagnosticTestResult.testIdentifier.id. ]  
( )

**[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 DiagnosticTestResults 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). ] (RS\_DEXT\_00073)

<b>Class</b>	<b>DiagnosticRequestOnBoardMonitoringTestResults</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x06_RequestOnBoardMonitoringTestResults
<b>Note</b>	This meta-class represents the ability to model an instance of the OBD mode 0x06 service. <b>Tags:</b> atp.recommendedPackage=DiagnosticRequestOnBoardMonitoringTestResultss
<b>Base</b>	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable





<b>Class</b>				
<b>DiagnosticRequestOnBoardMonitoringTestResults</b>				
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
diagnosticTestResult	<a href="#">DiagnosticTestResult</a>	*	ref	This reference identifies the applicable collection of test identifiers for setting up a request message for mode 0x06.
requestOnBoardMonitoringTestResultsClass	<a href="#">DiagnosticRequestOnBoardMonitoringTestResultsClass</a>	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 DiagnosticRequestOnBoardMonitoringTestResults in the given context.

**Table 5.106: DiagnosticRequestOnBoardMonitoringTestResults**

<b>Class</b>				
<b>DiagnosticRequestOnBoardMonitoringTestResultsClass</b>				
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x06_RequestOnBoardMonitoringTestResults			
<b>Note</b>	This meta-class represents the ability to define common properties for all instances of the "Request On-Board Monitoring Test Results" OBD diagnostic service.  <b>Tags:</b> atp.recommendedPackage=DiagnosticRequestOnBoardMonitoringTestResultss			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 5.107: DiagnosticRequestOnBoardMonitoringTestResultsClass**

### 5.6.6 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` | (RS\_DEXT\_00074)

The modeling is sketched in Figure 5.35.

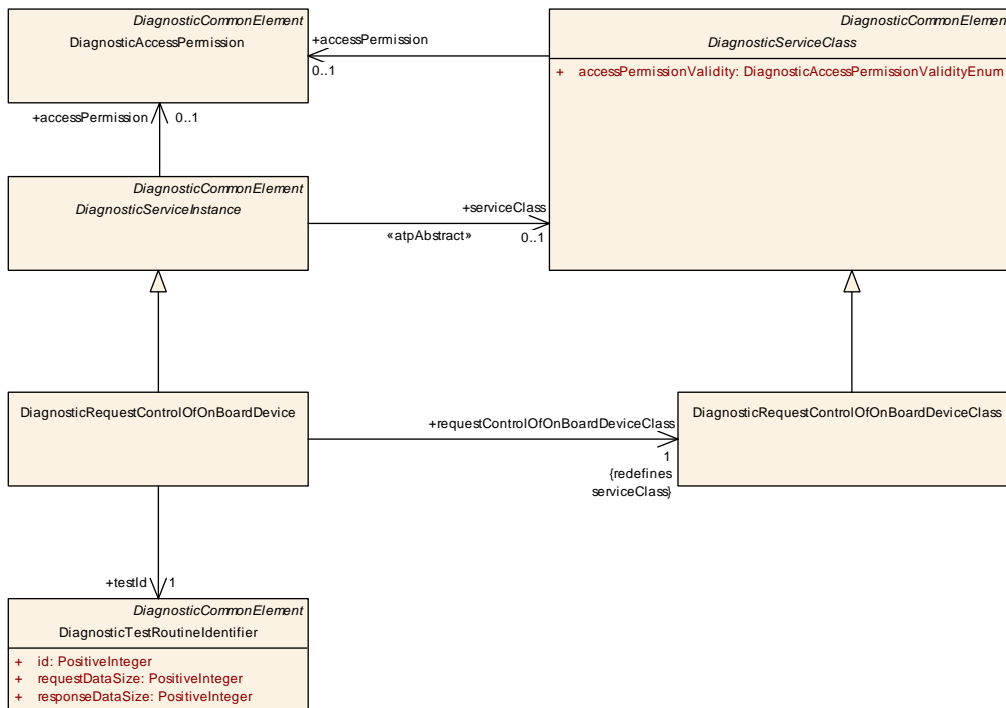


Figure 5.35: Modeling of diagnostic service for OBD Mode 0x08

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** [ The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of DiagnosticTestRoutineIdentifier.id. ]()

Class	DiagnosticRequestControlOfOnBoardDevice			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x08_RequestControlOfOnBoardDevice			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x08 service. <b>Tags:</b> atp.recommendedPackage=DiagnosticRequestControlOfOnBoardDevices			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	Type	Mul.	Kind	Note
requestControlOfOnBoardDeviceClass	DiagnosticRequestControlOfOnBoardDeviceClass	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.
testId	DiagnosticTestRoutineIdentifier	1	ref	This represents the test Id for the mode 0x08.

Table 5.108: DiagnosticRequestControlOfOnBoardDevice

<b>Class</b>	<b>DiagnosticRequestControlOfOnBoardDeviceClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x08_RequestControlOfOnBoard Device			
<b>Note</b>	This meta-class represents the ability to define common properties for all instances of the "Request Control Of On-Board Device" OBD diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticRequestControlOfOnBoardDevices			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 5.109: DiagnosticRequestControlOfOnBoardDeviceClass**

<b>Class</b>	<b>DiagnosticTestRoutineIdentifier</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x08_RequestControlOfOnBoard Device			
<b>Note</b>	This represents the test id of the DiagnosticTestIdentifier. <b>Tags:</b> atp.recommendedPackage=DiagnosticTestRoutineIdentifier			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">Multilanguage Referrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
id	PositiveInteger	1	attr	This represents the numerical id of the DiagnosticTest Identifier (see SAE J1979-DA).
requestData Size	PositiveInteger	1	attr	This represents the specified data size for the request message. Unit: byte.
responseData Size	PositiveInteger	1	attr	This represents the specified data size for the response message. Unit:byte.

**Table 5.110: DiagnosticTestRoutineIdentifier**

### 5.6.7 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`. ] ([RS\\_DEXT\\_00075](#))

The modeling is sketched in Figure [5.36](#).

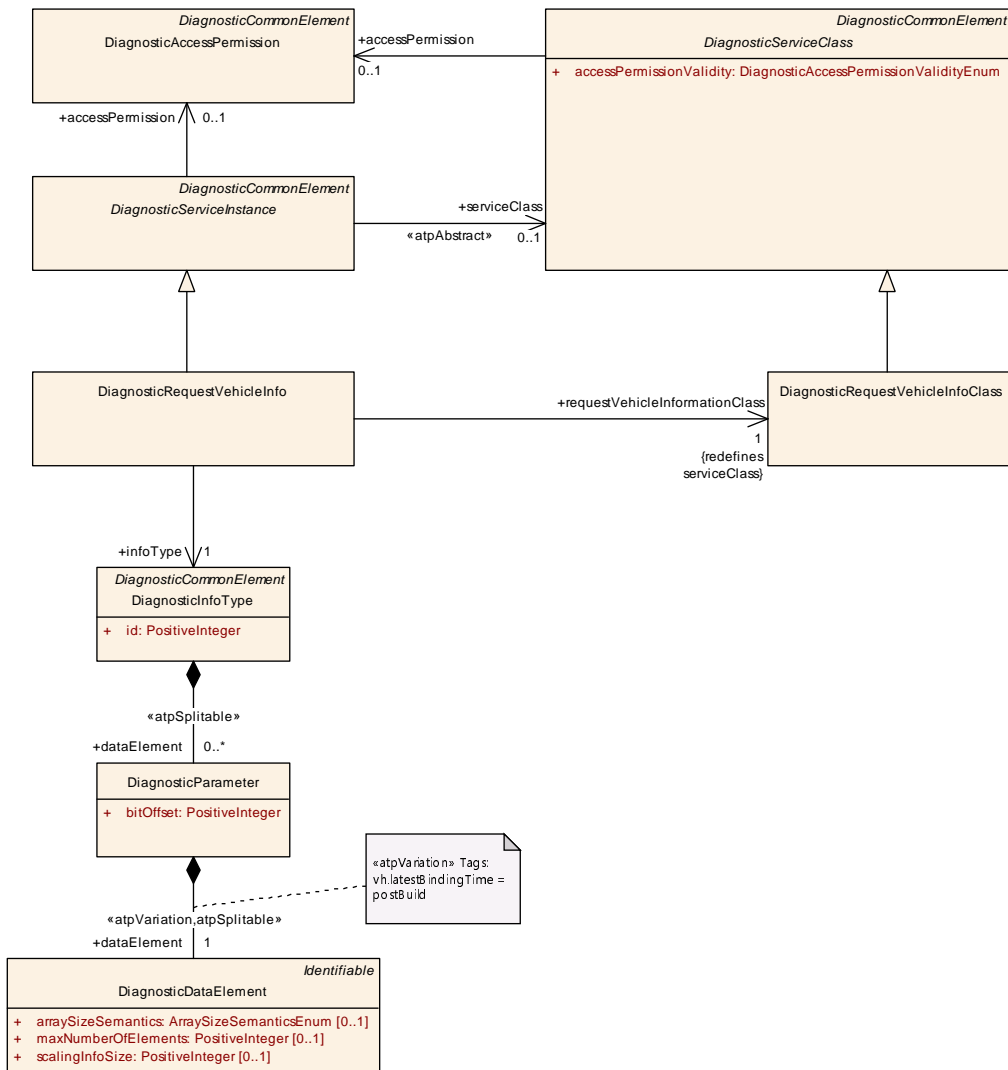


Figure 5.36: Modeling of diagnostic service for OBD Mode 0x09

**[constr\_1460] Restrictions for the value of `DiagnosticInfoType.id`** [ The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of `DiagnosticInfoType.id`. ]()



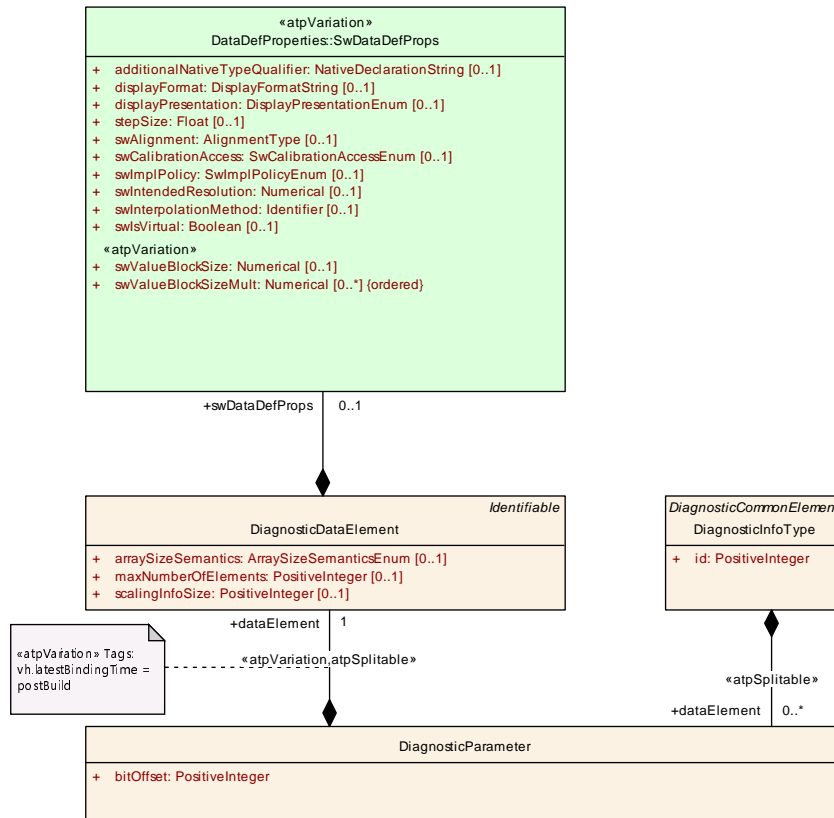


Figure 5.37: Modeling of DiagnosticInfoType

<b>Class</b>	<b>DiagnosticRequestVehicleInfo</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x09_RequestVehicleInformation			
<b>Note</b>	This meta-class represents the ability to model an instance of the OBD mode 0x09 service. <b>Tags:</b> atp.recommendedPackage=DiagnosticRequestVehicleInfos			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
infoType	<a href="#">DiagnosticInfoType</a>	1	ref	This represents the info type associated with the mode 0x09 service.
requestVehicleInformationClass	<a href="#">DiagnosticRequestVehicleInfoClass</a>	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 DiagnosticRequestVehicleInfo in the given context.

Table 5.111: DiagnosticRequestVehicleInfo

<b>Class</b>	<b>DiagnosticRequestVehicleInfoClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x09_RequestVehicleInformation			
<b>Note</b>	This meta-class represents the ability to define common properties for all instances of the "Request Vehicle Info" OBD diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticRequestVehicleInfos			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
-	-	-	-	-

**Table 5.112: DiagnosticRequestVehicleInfoClass**

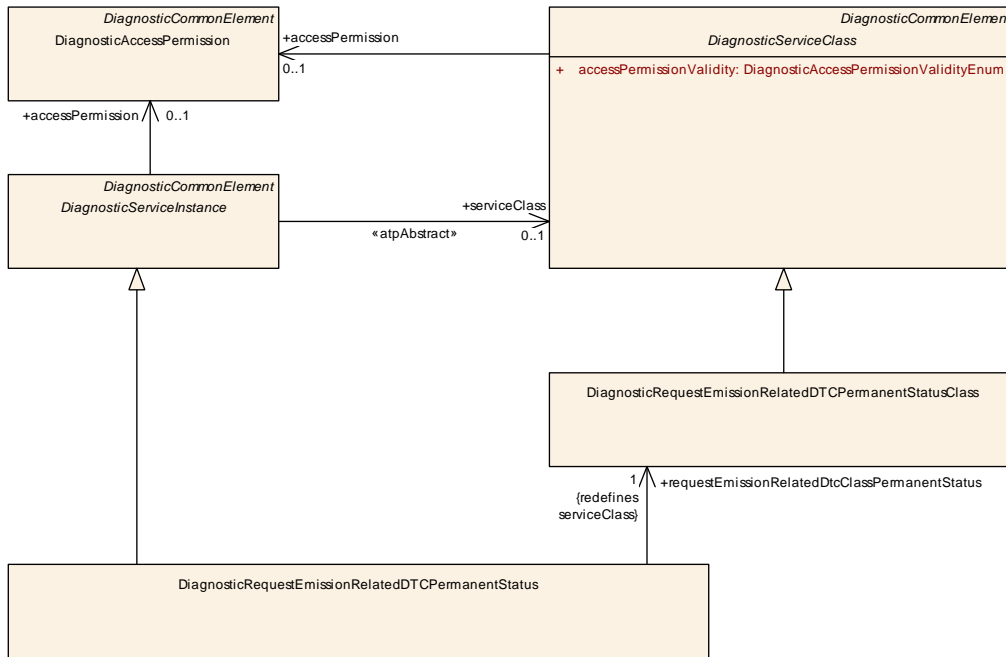
<b>Class</b>	<b>DiagnosticInfoType</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
<b>Note</b>	This meta-class represents the ability to model an OBD info type. <b>Tags:</b> atp.recommendedPackage=DiagnosticInfoTypes			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataElement	<a href="#">DiagnosticParameter</a>	*	aggr	This represents the data associated with the enclosing DiagnosticInfoType. <b>Stereotypes:</b> atp.Splittable <b>Tags:</b> atp.Splitkey=dataElement
id	PositiveInteger	1	attr	This attribute represents the value of InfoType (see SAE J1979-DA).

**Table 5.113: DiagnosticInfoType**

### 5.6.8 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`. ] ([RS\\_DEXT\\_00076](#))

The modeling is sketched in Figure [5.38](#).



**Figure 5.38: Modeling of diagnostic service for OBD Modes 0x0A**

<b>Class</b>	<b>DiagnosticRequestEmissionRelatedDTCPermanentStatus</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x0A_RequestEmissionRelatedDTCPermanentStatus			
<b>Note</b>	This meta-class represents the ability to model an instance of the OBD mode 0x0A service. <b>Tags:</b> atp.recommendedPackage=DiagnosticRequestEmissionRelatedDTCPermanentStatuss			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceInstance</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
requestEmissionRelatedDtcClassPermanentStatus	<a href="#">DiagnosticRequestEmissionRelatedDTCPermanentStatusClass</a>	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 DiagnosticRequestEmissionRelatedDTCPermanentStatus in the given context.

**Table 5.114: DiagnosticRequestEmissionRelatedDTCPermanentStatus**

<b>Class</b>	<b>DiagnosticRequestEmissionRelatedDTCPermanentStatusClass</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x0A_RequestEmissionRelatedDTCPermanentStatus			
<b>Note</b>	This meta-class represents the ability to define common properties for all instances of the "Request Emission Related DTC Permanent Status" OBD diagnostic service. <b>Tags:</b> atp.recommendedPackage=DiagnosticRequestEmissionRelatedDTCPermanentStatuss			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticServiceClass</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
-	-	-	-	-

**Table 5.115: DiagnosticRequestEmissionRelatedDTCPermanentStatusClass**

## 5.7 UDS Diagnostic Services for supporting WWH-OBD

[TPS\_DEXT\_01133] **Support for WWH-OBD within the diagnostic extract** [ A support for WWH-OBD [19] 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)

](*RS\_DEXT\_00077*)

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 ISO 26021 [17]) the attribute has no semantics.

## 5.8 Diagnostic Service Mapping

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 case where the `DiagnosticExtract` refers to software-components in general and `PortPrototypes` in the context of either `CompositionSwComponentTypes` or `AtomicSwComponentTypes`:

- `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`.

|(RS\_DEXT\_00052)

<b>Class</b>	<b>SwcServiceDependency</b>			
<b>Package</b>	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ServiceMapping			
<b>Note</b>	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.			
<b>Base</b>	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable, ServiceDependency			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
assignedData	RoleBasedData Assignment	*	aggr	Defines the role of an associated data object of the same component. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime
assignedPort	RoleBasedPort Assignment	*	aggr	Defines the role of an associated port of the same component. <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=assignedPort, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
representedPort Group	PortGroup	0..1	ref	This reference specifies an association between the ServiceNeeds 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.
serviceNeeds	ServiceNeeds	1	aggr	The associated ServiceNeeds.

**Table 5.116: SwcServiceDependency**

<b>Class</b>	<b>BswServiceDependency</b>			
<b>Package</b>	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
<b>Note</b>	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.			
<b>Base</b>	ARObject, ServiceDependency			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
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. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime
assignedEntry Role	RoleBasedBswModule EntryAssignment	*	aggr	Defines the role of an associated BswModuleEntry in the context of the ServiceNeeds element. <b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=assignedEntryRole, variation Point.shortLabel vh.latestBindingTime=preCompileTime
ident	<a href="#">BswService DependencyIdent</a>	0..1	aggr	This adds the ability to become referable to BswService Dependency. <b>Tags:</b> atp.Status=shallBecomeMandatory xml.sequenceOffset=-100
serviceNeeds	<a href="#">ServiceNeeds</a>	1	aggr	The associated ServiceNeeds.

**Table 5.117: BswServiceDependency**

**[constr\_1450] Service mapping for ODB mode 0x01 for [DiagnosticParameterIdentifier](#)** [ if a [DiagnosticServiceSwMapping](#) or [DiagnosticServiceDataMapping](#) refers to a [DiagnosticRequestCurrentPowertrainData](#) and a [DiagnosticDataElement](#) that is aggregated by a [DiagnosticParameterIdentifier](#) then 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](#).

]()

**[constr\_1451] Service mapping for OBD mode 0x09 for [DiagnosticInfoType](#)** [ if a [DiagnosticServiceSwMapping](#) refers to [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** [ if a **DiagnosticServiceSwMapping** refers to a **DiagnosticRequestControlOfOnBoardDevice** then the **SwcServiceDependency** referenced by the same **DiagnosticServiceSwMapping** shall aggregate an **ObdControlServiceNeeds** in the role **serviceNeeds**. ]()

<b>Class</b>	<b>ObdInfoServiceNeeds</b>			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
<b>Note</b>	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.			
<b>Base</b>	<i>ARObject</i> , <i>DiagnosticCapabilityElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> , <i>Service Needs</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataLength	PositiveInteger	0..1	attr	This attribute is applicable only if the ServiceNeeds is aggregated within BswModuleDependency. This attribute represents the length of data (in bytes) provided for this InfoType.
infoType	PositiveInteger	0..1	attr	The InfoType according to ISO 15031-5

**Table 5.118: ObdInfoServiceNeeds**

<b>Class</b>	<b>ObdPidServiceNeeds</b>			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
<b>Note</b>	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).			
<b>Base</b>	<i>ARObject</i> , <i>DiagnosticCapabilityElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> , <i>Service Needs</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataLength	PositiveInteger	0..1	attr	This attribute is applicable only if the ServiceNeeds is aggregated within BswModuleDependency. This attribute represents the length of data (in bytes) provided for this particular PID signal.
parameterId	PositiveInteger	0..1	attr	Standardized parameter identifier (PID) according to the OBD standard specified in attribute "standard".
standard	String	0..1	attr	Annotates the standard according to which the PID is given, e.g. "ISO15031-5" or "SAE J1979 Rev May 2007".

**Table 5.119: ObdPidServiceNeeds**

<b>Class</b>	<b>ObdControlServiceNeeds</b>			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
<b>Note</b>	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.			
<b>Base</b>	<i>ARObject</i> , <i>DiagnosticCapabilityElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> , <i>Service Needs</i>			



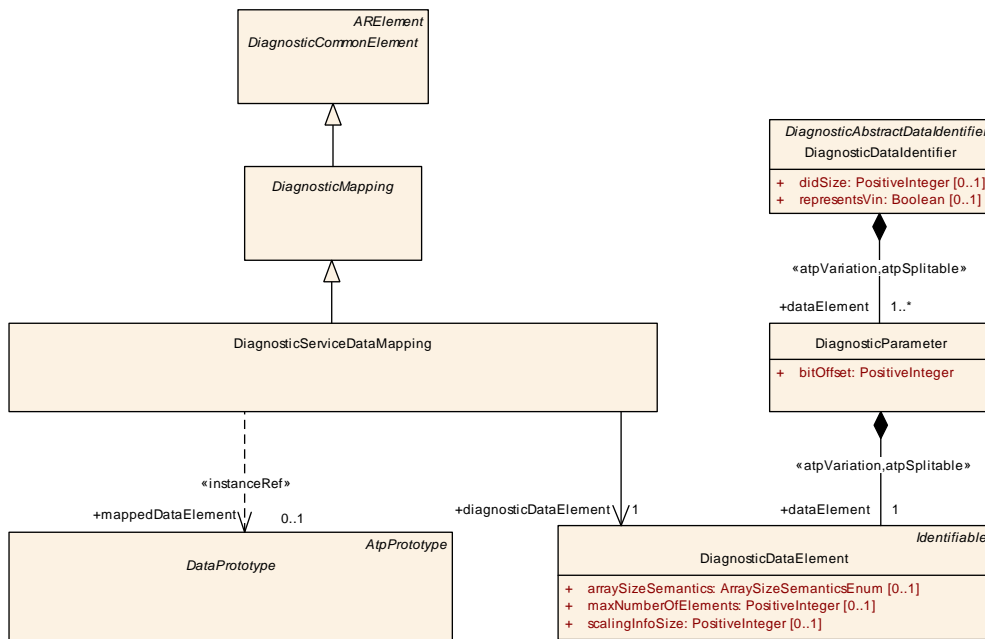


Class	ObdControlServiceNeeds			
Attribute	Type	Mul.	Kind	Note
testId	PositiveInteger	0..1	attr	Test Identifier (TID) according to ISO 15031-5.

**Table 5.120: ObdControlServiceNeeds**

### 5.8.1 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 5.39: Dcm service data mapping**

If several `DiagnosticServiceDataMappings` exist that refer to the context of the same `VariableDataPrototype` and different `DiagnosticDataElements` aggregated by the same `DiagnosticDataIdentifier` then this is a strong indication that an atomic access to a structured data type is intended.

The association of elements of the data structure to `DiagnosticDataElements` is implicitly given by means of matching `shortName` or by explicit references into the internals of the `VariableDataPrototype`.

Please note that this approach of associating elements of a composite data structure to `DiagnosticDataElements` of one `DiagnosticDataIdentifier` based on the `shortName` can only work if the depth of the composite data structure is limited to 1.



**[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`.

This type of data access is suitable for the diagnostic services `ReadDataByIdentifier (0x22)`, `WriteDataByIdentifier (0x2E)`, and `InputOutputControl (0x2F)`. ]([RS\\_DEXT\\_00052](#))

<b>Class</b>	<b>DiagnosticServiceDataMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::ServiceMapping			
<b>Note</b>	<p>This represents the ability to define a mapping of a diagnostic service to a software-component.</p> <p>This kind of service mapping is applicable for the usage of <code>SenderReceiverInterfaces</code> or event/notifier semantics in <code>ServiceInterfaces</code> on the adaptive platform.</p> <p><b>Tags:</b> atp.recommendedPackage=DiagnosticServiceMappings</p>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMapping</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
diagnosticDataElement	<a href="#">DiagnosticDataElement</a>	1	ref	This represents the applicable payload that corresponds to the referenced <code>DataPrototype</code> in the role <code>mappedDataElement</code> or (in case of a usage on the adaptive platform) <code>mappedApDataElement</code> .
mappedDataElement	<a href="#">DataPrototype</a>	0..1	iref	This represents the <code>dataElement</code> in the application software that is accessed for diagnostic purpose. This role is applicable on the classic platform.

**Table 5.121: DiagnosticServiceDataMapping**

**[constr\_1343] Simultaneous existence of the attributes `DiagnosticServiceDataMapping.diagnosticDataElement` and `DiagnosticDataByIdentifier.dataIdentifier`** [ A `DiagnosticServiceDataMapping.diagnosticDataElement` shall also be aggregated by a `DiagnosticDataByIdentifier` in the role `dataIdentifier.dataElement.dataElement`. ]()

Please note that [\[constr\\_1343\]](#) shall only apply for the step in the methodology where the `DiagnosticExtract` is considered complete to the point that the configuration of the Dcm and Dem can be derived. Any intermediate step, e.g. hand-over from OEM to tier-1 supplier does not actually enforce [\[constr\\_1343\]](#).

In other words, [\[constr\\_1343\]](#) makes sure that there is a connection between the `DiagnosticServiceDataMapping` and the corresponding `DiagnosticReadDataByIdentifier` or `DiagnosticWriteDataByIdentifier`.

Only by this means the diagnostic service becomes fully usable.

**[constr\_1344] Condition for the identification of data types of attributes `DiagnosticServiceDataMapping.mappedDataElement` and `DiagnosticServiceDataMapping.diagnosticDataElement`** [ Both `DiagnosticServiceDataMapping.mappedDataElement` and `DiagnosticServiceDataMapping.diagnosticDataElement` shall be typed by either of the following options:

- [ApplicationPrimitiveDataType](#) where the value of attribute `category` is set to `VALUE`.
- [ImplementationDataType](#) where the value of attribute `category` is set to `VALUE` or to `TYPE_REFERENCE` that eventually resolves to an [ImplementationDataType](#) where attribute `category` is set to `VALUE`.

]()

<b>Class</b>	<b>ApplicationPrimitiveDataType</b>			
<b>Package</b>	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
<b>Note</b>	A primitive data type defines a set of allowed values. <b>Tags:</b> atp.recommendedPackage=ApplicationDataTypes			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">ApplicationDataType</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpType</a> , <a href="#">AutosarDataType</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
-	-	-	-	-

**Table 5.122: ApplicationPrimitiveDataType**

<b>Class</b>	<b>ImplementationDataType</b>			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::ImplementationDataTypes			
<b>Note</b>	Describes a reusable data type on the implementation level. This will typically correspond to a typedef in C-code. <b>Tags:</b> atp.recommendedPackage=ImplementationDataTypes			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AbstractImplementationDataType</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpType</a> , <a href="#">AutosarDataType</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dynamicArray SizeProfile	String	0..1	attr	Specifies the profile which the array will follow in case this data type is a variable size array.
isStructWith Optional Element	Boolean	0..1	attr	This attribute is only valid if the attribute category is set to <code>STRUCTURE</code> .  If set to <code>True</code> , this attribute indicates that the <code>ImplementationDataType</code> has been created with the intention to define at least one element of the structure as optional. <b>Tags:</b> atp.Status=draft
subElement (or- dered)	ImplementationData TypeElement	*	aggr	Specifies an element of an array, struct, or union data type.  The aggregation of <code>ImplementationDataTypeElement</code> is subject to variability with the purpose to support the conditional existence of elements inside a <code>ImplementationDataType</code> representing a structure. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the <code>SymbolProps</code> for the <code>ImplementationDataType</code> . <b>Stereotypes:</b> atpSplittable <b>Tags:</b> atp.Splitkey=shortName





Class	ImplementationDataType			
typeEmitter	NameToken	0..1	attr	This attribute is used to control which part of the AUTOSAR toolchain is supposed to trigger data type definitions.

**Table 5.123: ImplementationDataType**

In other words, [constr\_1344] requires that both `DiagnosticServiceDataMapping.mappedDataElement`<sup>4</sup> and `DiagnosticServiceDataMapping.diagnosticDataElement` shall be typed by a primitive data type.

Please refer to [8] 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** [ A `DiagnosticDataElement` that is referenced by a `DiagnosticServiceDataMapping` shall not (finally) be aggregated by a `DiagnosticRoutine`. ]()

**[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 `dataElements` in an `ApplicationSwComponentType`.

Therefore, [constr\_1345] does intentionally not exclude the aggregation of `DiagnosticDataElement` by `DiagnosticExtendedDataRecord` in the context of `DiagnosticServiceDataMapping`. ](*RS\_DEXT\_00052*)

Class	ApplicationSwComponentType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	The <code>ApplicationSwComponentType</code> is used to represent the application software. <b>Tags:</b> atp.recommendedPackage=SwComponentTypes			
Base	<code>ARElement</code> , <code>ARObject</code> , <code>AtomicSwComponentType</code> , <code>AtpBlueprint</code> , <code>AtpBlueprintable</code> , <code>AtpClassifier</code> , <code>AtpType</code> , <code>CollectableElement</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>PackageableElement</code> , <code>Referrable</code> , <code>SwComponentType</code>			
Attribute	Type	Mul.	Kind	Note
-	-	-	-	-

**Table 5.124: ApplicationSwComponentType**

**[constr\_1584] DiagnosticDataElement shall not be used more than once in I/O Control instance** [ 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`. ]()

<sup>4</sup>`DiagnosticServiceDataMapping.mappedDataElement` can be an element of a (potentially large) composite data type. The utility of this is that this way the footprint of the data access to the payload of request and response messages can be kept as low as possible.

## 5.8.2 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).

<b>Class</b>	<b>ClientServerInterface</b>			
<b>Package</b>	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
<b>Note</b>	A client/server interface declares a number of operations that can be invoked on a server by a client. <b>Tags:</b> atp.recommendedPackage=PortInterfaces			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpType</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">PortInterface</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
operation	ClientServerOperation	1..*	aggr	ClientServerOperation(s) of this ClientServerInterface. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=blueprintDerivationTime
possibleError	ApplicationError	*	aggr	Application errors that are defined as part of this interface.

**Table 5.125: ClientServerInterface**

The following constraint only applies to the case where I/O Control is implemented via sender-receiver-communication (see [TPS\_SWCT\_01654])

**[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 to the [SwcServiceDependency](#) (if the service applies to the application software) or [BswServiceDependency](#) (if the service applies to the basic software). ] ([RS\\_DEXT\\_00052](#))

It is required to use the applicable form of reference to the target [SwcServiceDependency](#) depending on the context of the enclosing [AtomicSwComponentType](#).

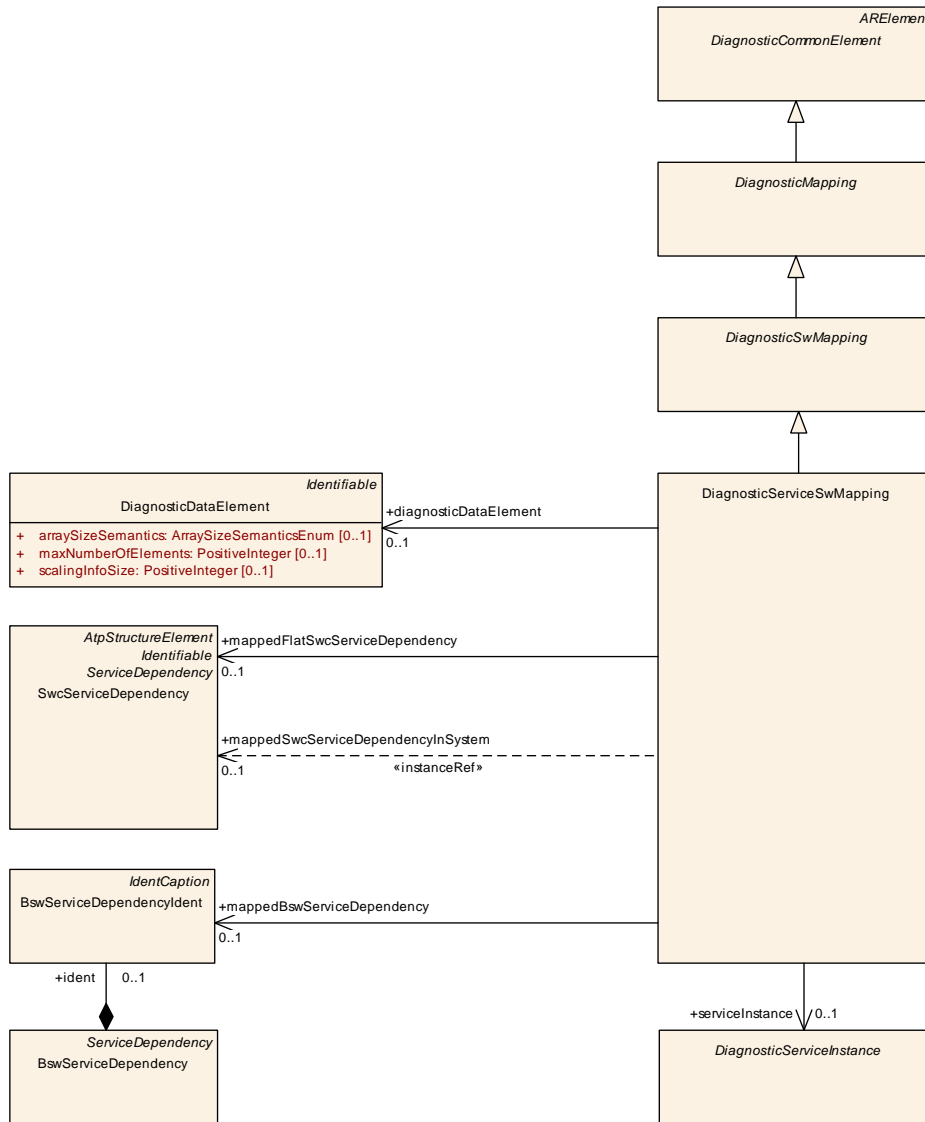


Figure 5.40: Dcm service software mapping

<b>Class</b>	<i>DiagnosticSwMapping</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::ServiceMapping			
<b>Note</b>	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.			
<b>Base</b>	<i>ARElement</i> , <i>ARObject</i> , <i>CollectableElement</i> , <i>DiagnosticCommonElement</i> , <i>DiagnosticMapping</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
<b>Subclasses</b>	<i>DiagnosticEnableConditionPortMapping</i> , <i>DiagnosticEventPortMapping</i> , <i>DiagnosticFimFunctionMapping</i> , <i>DiagnosticOperationCyclePortMapping</i> , <i>DiagnosticServiceSwMapping</i> , <i>DiagnosticStorageConditionPortMapping</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
—	—	—	—	—

Table 5.126: DiagnosticSwMapping

<b>Class</b>	<b>DiagnosticServiceSwMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::ServiceMapping			
<b>Note</b>	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. <b>Tags:</b> atp.recommendedPackage=DiagnosticServiceMappings			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMapping</a> , <a href="#">DiagnosticSwMapping</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
diagnosticDataElement	<a href="#">DiagnosticDataElement</a>	0..1	ref	This represents a <a href="#">DiagnosticDataElement</a> required to execute the respective diagnostic service in the context of the diagnostic service mapping,
mappedBswServiceDependency	<a href="#">BswServiceDependencyIdent</a>	0..1	ref	This is supposed to represent a reference to a <a href="#">BswServiceDependency</a> . the latter is not derived from <a href="#">Referrable</a> and therefore this detour needs to be implemented to still let <a href="#">BswServiceDependency</a> become the target of a reference.
mappedFlatSwcServiceDependency	<a href="#">SwcServiceDependency</a>	0..1	ref	This represents the ability to refer to an <a href="#">AtomicSwComponentType</a> that is available without the definition of how it will be embedded into the component hierarchy.
mappedSwcServiceDependencyInSystem	<a href="#">SwcServiceDependency</a>	0..1	iref	This represents the ability to point into the component hierarchy (under possible consideration of the root <a href="#">SoftwareComposition</a> )
serviceInstance	<a href="#">DiagnosticServiceInstance</a>	0..1	ref	This represents the service instance that needs to be considered in this diagnostics service mapping.

**Table 5.127: 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. |(RS\_DEXT\_00052)

Please find further details in Figure 5.40.

Please note that the introduction [TPS\_DEXT\_01044], although being dangerously close to a hack, 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.

<b>Class</b>	<b>BswServiceDependencyIdent</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::ServiceMapping			
<b>Note</b>	This meta-class is created to add the ability to become the target of a reference to the non-Referrable <code>BswServiceDependency</code> .			
<b>Base</b>	<code>ARObject</code> , <code>AtpClassifier</code> , <code>AtpFeature</code> , <code>AtpStructureElement</code> , <code>IdentCaption</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 5.128: BswServiceDependencyIdent**

**[constr\_1346] Allowed values of `DiagnosticServiceSwMapping.serviceInstance`** [ The applicability of the `DiagnosticServiceSwMapping` is limited to pre-defined 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 attributes of `DiagnosticServiceSwMapping`** [ 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 the fact that at most a single `SwcServiceDependency` can be referenced by a `DiagnosticServiceSwMapping` and this `SwcServiceDependency` cannot be identified by both `mappedSwcServiceDependencyInSystem` and `mappedFlatSwcServiceDependency`.

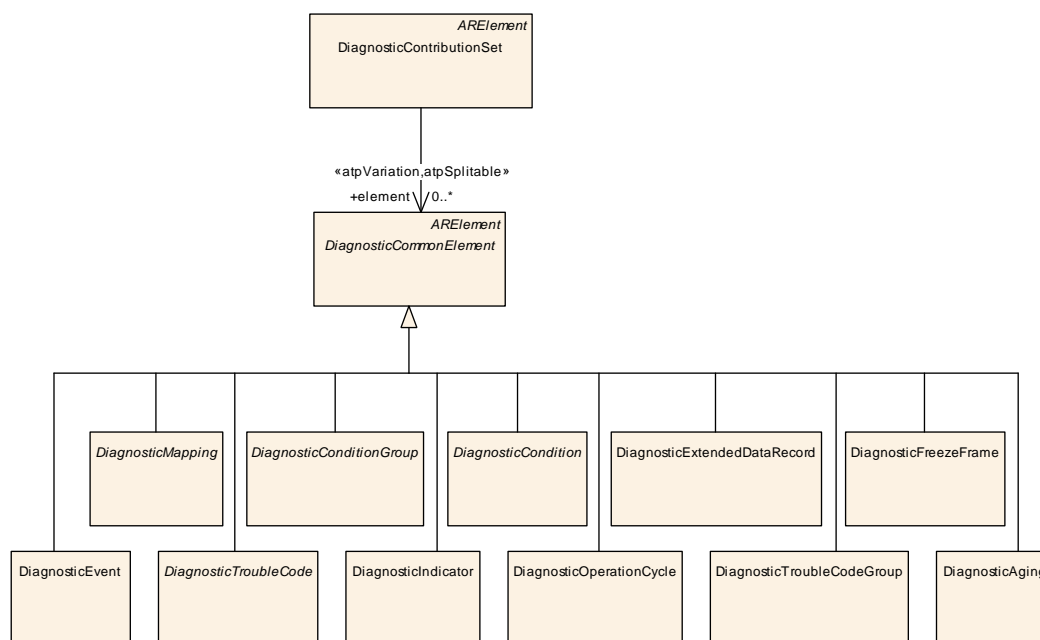
## 6 Diagnostic Event Handling

### 6.1 Introduction

This subchapter describes the meta-model elements that define the handling of and the functionality around diagnostic events.

In a standard AUTOSAR Basic Software architecture, the definitions based on the model elements described in this subchapter 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 6.1: 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.

### 6.2 DiagnosticEvent

**[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. ] ([RS\\_DEXT\\_00023](#))

Figure 6.2 depicts the definition of `DiagnosticEvent` together with its properties.



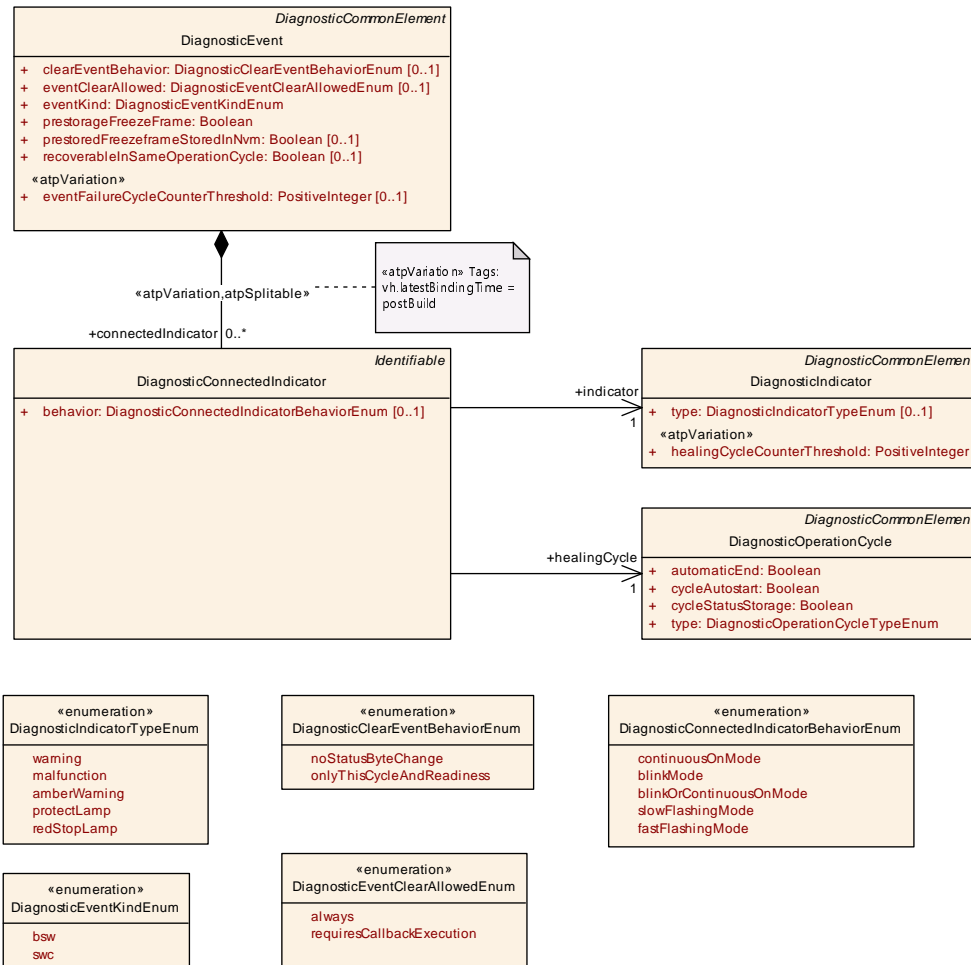


Figure 6.2: Modeling of DiagnosticEvent

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 handling of BSW events (e.g. definition of associated `DiagnosticTroubleCode`).

[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 on this callback interface can be expressed using the attribute `eventClearAllowed`:

- `always` indicates that a clearing request for the `DiagnosticEvent` shall unconditionally be executed.
- In 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`.

|(RS\_DEXT\_00023)

Class	DiagnosticEvent			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
Note	This element is used to configure DiagnosticEvents. <b>Tags:</b> atp.recommendedPackage=DiagnosticEvents			
Base	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
Attribute	Type	Mul.	Kind	Note
clearEvent Behavior	<a href="#">DiagnosticClearEventBehaviorEnum</a>	0..1	attr	This attribute defines the resulting UDS status byte for the related event, which shall not be cleared according to the <code>ClearEventAllowed</code> callback.
connected Indicator	<a href="#">DiagnosticConnectedIndicator</a>	*	aggr	Event specific description of Indicators. <b>Stereotypes:</b> atp.Splittable; atp.Variation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
eventClear Allowed	<a href="#">DiagnosticEventClearAllowedEnum</a>	0..1	attr	This attribute defines whether the Dem has access to a "ClearEventAllowed" callback.
eventFailure CycleCounter Threshold	PositiveInteger	0..1	attr	This attribute defines the number of failure cycles for the event based fault confirmation. <b>Stereotypes:</b> atp.Variation <b>Tags:</b> vh.latestBindingTime=postBuild
eventKind	<a href="#">DiagnosticEventKindEnum</a>	1	attr	This attribute is used to distinguish between SWC and BSW events.
prestorage FreezeFrame	Boolean	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 False: Prestorage of FreezeFrames is not supported
prestored FreezeFrame StoredInNvm	Boolean	0..1	attr	If the Event uses a prestored freeze-frame (using the operations <code>PrestoreFreezeFrame</code> and <code>ClearPrestoredFreezeFrame</code> of the service interface <code>DiagnosticMonitor</code> ) 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 6.1: DiagnosticEvent**

<b>Class</b>	<b>RoleBasedPortAssignment</b>			
<b>Package</b>	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ServiceMapping			
<b>Note</b>	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.			
<b>Base</b>	ARObject			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
portPrototype	PortPrototype	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.
role	Identifier	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 6.2: 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. ] (*RS\_DEXT\_00023*)

[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*. ] (*RS\_DEXT\_00023, RS\_DEXT\_00045*)

[TPS\_DEXT\_01068] **Textual description with respect to the DiagnosticEvent** [ 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*. ] (*RS\_DEXT\_00023, RS\_DEXT\_00045*)

<b>Class</b>	<b>StructuredReq</b>			
<b>Package</b>	M2::MSR::Documentation::BlockElements::RequirementsTracing			
<b>Note</b>	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.			
<b>Base</b>	ARObject, <i>DocumentViewSelectable</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Paginateable</i> , <i>Referrable</i> , <i>Traceable</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
appliesTo	standardNameEnum	*	attr	This attribute represents the platform the requirement is assigned to.  <b>Tags:</b> xml.namePlural=APPLIES-TO-DEPENDENCIES xml.sequenceOffset=25
conflicts	<i>DocumentationBlock</i>	0..1	aggr	This represents an informal specification of conflicts.  <b>Tags:</b> xml.sequenceOffset=40





Class	StructuredReq			
date	DateTime	1	attr	This represents the date when the requirement was initiated. <b>Tags:</b> 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. <b>Tags:</b> xml.sequenceOffset=30
description	DocumentationBlock	0..1	aggr	This represents the general description of the requirement. <b>Tags:</b> xml.sequenceOffset=10
importance	String	1	attr	This allows to represent the importance of the requirement. <b>Tags:</b> xml.sequenceOffset=8
issuedBy	String	1	attr	This represents the person, organization or authority which issued the requirement. <b>Tags:</b> xml.sequenceOffset=6
rationale	DocumentationBlock	0..1	aggr	This represents the rationale of the requirement. <b>Tags:</b> 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. <b>Tags:</b> xml.sequenceOffset=60
supporting Material	DocumentationBlock	0..1	aggr	This represents an informal specification of the supporting material. <b>Tags:</b> 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. <b>Tags:</b> xml.sequenceOffset=70
type	String	1	attr	This attribute allows to denote the type of requirement to denote for example is it an "enhancement", "new feature" etc. <b>Tags:</b> 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. <b>Tags:</b> xml.sequenceOffset=35

**Table 6.3: StructuredReq**

For more details regarding the modeling of the semi-formal text please refer to Figure 4.3.

[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 must exist.
- **DIAG\_EVENT\_DEMAT\_TIME**: this value describes the de-mature time for the corresponding [DiagnosticEvent](#), i.e. how long or how often must the OK conditions be fulfilled.

|(RS\_DEXT\_00001, RS\_DEXT\_00023, RS\_DEXT\_00045)

<b>Class</b>	<b>DiagnosticConnectedIndicator</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
<b>Note</b>	Description of indicators that are defined per DiagnosticEvent.			
<b>Base</b>	ARObject, <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
behavior	<a href="#">DiagnosticConnectedIndicatorBehaviorEnum</a>	0..1	attr	Behavior of the linked indicator.
healingCycle	<a href="#">DiagnosticOperationCycle</a>	1	ref	The deactivation of indicators per event is defined as healing of a diagnostic event. The operation cycle in which the warning indicator will be switched off is defined here.
indicator	<a href="#">DiagnosticIndicator</a>	1	ref	Reference to the used indicator.

**Table 6.4: DiagnosticConnectedIndicator**

<b>Enumeration</b>	<b>DiagnosticEventClearAllowedEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent
<b>Note</b>	Denotes whether clearing of events is allowed.
<b>Literal</b>	<b>Description</b>
always	The clearing is allowed unconditionally. <b>Tags:</b> atp.EnumerationValue=0





<i>Enumeration</i>	<b>DiagnosticEventClearAllowedEnum</b>
requiresCallbackExecution	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). <b>Tags:</b> atp.EnumerationValue=2

**Table 6.5: DiagnosticEventClearAllowedEnum**

<i>Enumeration</i>	<b>DiagnosticClearEventBehaviorEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent
<b>Note</b>	Possible behavior for clearing events.
<b>Literal</b>	<b>Description</b>
noStatusByteChange	The event status byte keeps unchanged. <b>Tags:</b> atp.EnumerationValue=0
onlyThisCycleAndReadiness	The OperationCycle and readiness bits of the event status byte are reset. <b>Tags:</b> atp.EnumerationValue=1

**Table 6.6: DiagnosticClearEventBehaviorEnum**

<i>Enumeration</i>	<b>DiagnosticEventKindEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent
<b>Note</b>	Applicability of the diagnostic event.
<b>Literal</b>	<b>Description</b>
bsw	The event is assigned to a BSW module. <b>Tags:</b> atp.EnumerationValue=0
swc	The event is assigned to a SWC. <b>Tags:</b> atp.EnumerationValue=1

**Table 6.7: DiagnosticEventKindEnum**

<i>Enumeration</i>	<b>DiagnosticConnectedIndicatorBehaviorEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent
<b>Note</b>	Behavior of the indicator.
<b>Literal</b>	<b>Description</b>
blinkMode	The indicator blinks when the event has status FAILED. <b>Tags:</b> atp.EnumerationValue=0
blinkOrContinuousOnMode	The indicator is active and blinks when the event has status FAILED. <b>Tags:</b> atp.EnumerationValue=1
continuousOnMode	The indicator is active when the event has status FAILED. <b>Tags:</b> atp.EnumerationValue=2
fastFlashingMode	Flash Indicator Lamp should be set to "Fast Flash". <b>Tags:</b> atp.EnumerationValue=3
slowFlashingMode	Flash Indicator Lamp should be set to "Slow Flash". <b>Tags:</b> atp.EnumerationValue=4

**Table 6.8: DiagnosticConnectedIndicatorBehaviorEnum**

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 `DiagnosticEvent`.

**Listing 6.1: Example for the definition of a semi-formal textual elements in the context of a `DiagnosticEvent`**

```

<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>DEMATURE_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>
  <CLEAR-EVENT-BEHAVIOR>ONLY-THIS-CYCLE-AND-READINESS</CLEAR-EVENT-
    BEHAVIOR>
  <EVENT-CLEAR-ALLOWED>ALWAYS</EVENT-CLEAR-ALLOWED>
  <EVENT-FAILURE-CYCLE-COUNTER-THRESHOLD>100</EVENT-FAILURE-CYCLE-COUNTER
    -THRESHOLD>
  <EVENT-KIND>SWC</EVENT-KIND>
  <PRESTORAGE-FREEZE-FRAME>>false</PRESTORAGE-FREEZE-FRAME>
</DIAGNOSTIC-EVENT>
  
```

### 6.3 DiagnosticTroubleCode

`DiagnosticTroubleCodes` (i.e. the ECU external view on diagnostic events) are defined together with their properties and mapped to `DiagnosticEvents` using `DiagnosticEventToTroubleCodeUdsMapping`.

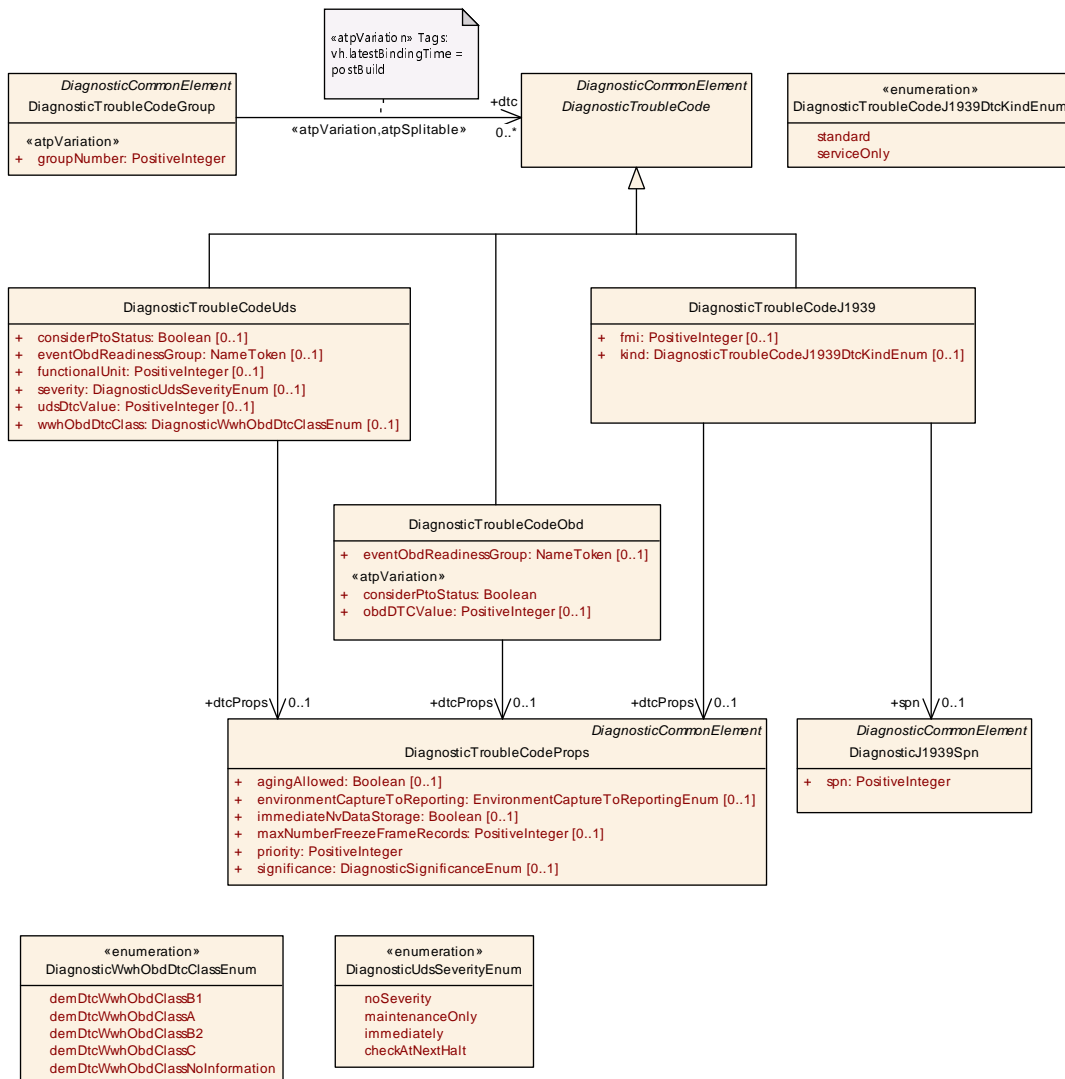


Figure 6.3: Modeling of **DiagnosticTroubleCode**

[TPS\_DEXT\_03012] **Three kinds of DTCs** [ There are three kinds of DTCs represented as specializations of **DiagnosticTroubleCode**:

- non OBD relevant DTCs (**DiagnosticTroubleCodeUds**)
- OBD relevant DTCs (**DiagnosticTroubleCodeObd**)
- J1939 [20] relevant DTCs (**DiagnosticTroubleCodeJ1939**)

Properties individual to such a DTC specialization are modeled as attributes of **DiagnosticTroubleCodeUds**, **DiagnosticTroubleCodeObd** and **DiagnosticTroubleCodeJ1939**, respectively. ] (**RS\_DEXT\_00024**)



<b>Class</b>	<b>DiagnosticTroubleCodeUds</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
<b>Note</b>	This element is used to describe non OBD-relevant DTCs. <b>Tags:</b> atp.recommendedPackage=DiagnosticTroubleCodes			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticTroubleCode</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
considerPtoStatus	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.
dtcProps	<a href="#">DiagnosticTroubleCodeProps</a>	0..1	ref	Defined properties associated with the DemDTC.
eventObdReadinessGroup	<a href="#">NameToken</a>	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.
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.
severity	<a href="#">DiagnosticUdsSeverityEnum</a>	0..1	attr	DTC severity according to ISO 14229-1.
udsDtcValue	PositiveInteger	0..1	attr	Unique Diagnostic Trouble Code value for UDS.
wwhObdDtcClass	<a href="#">DiagnosticWwhObdDtcClassEnum</a>	0..1	attr	This attribute is used to identify (if applicable) the corresponding severity class of an WWH-OBD DTC.

**Table 6.9: DiagnosticTroubleCodeUds**

<b>Class</b>	<b>DiagnosticTroubleCodeObd</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
<b>Note</b>	This element is used to define OBD-relevant DTCs. <b>Tags:</b> atp.recommendedPackage=DiagnosticTroubleCodes			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticTroubleCode</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
considerPtoStatus	Boolean	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. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime
dtcProps	<a href="#">DiagnosticTroubleCodeProps</a>	0..1	ref	Defined properties associated with the DemDTC.
eventObdReadinessGroup	<a href="#">NameToken</a>	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.
obdDTCValue	PositiveInteger	0..1	attr	Unique Diagnostic Trouble Code value for OBD. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime

**Table 6.10: DiagnosticTroubleCodeObd**

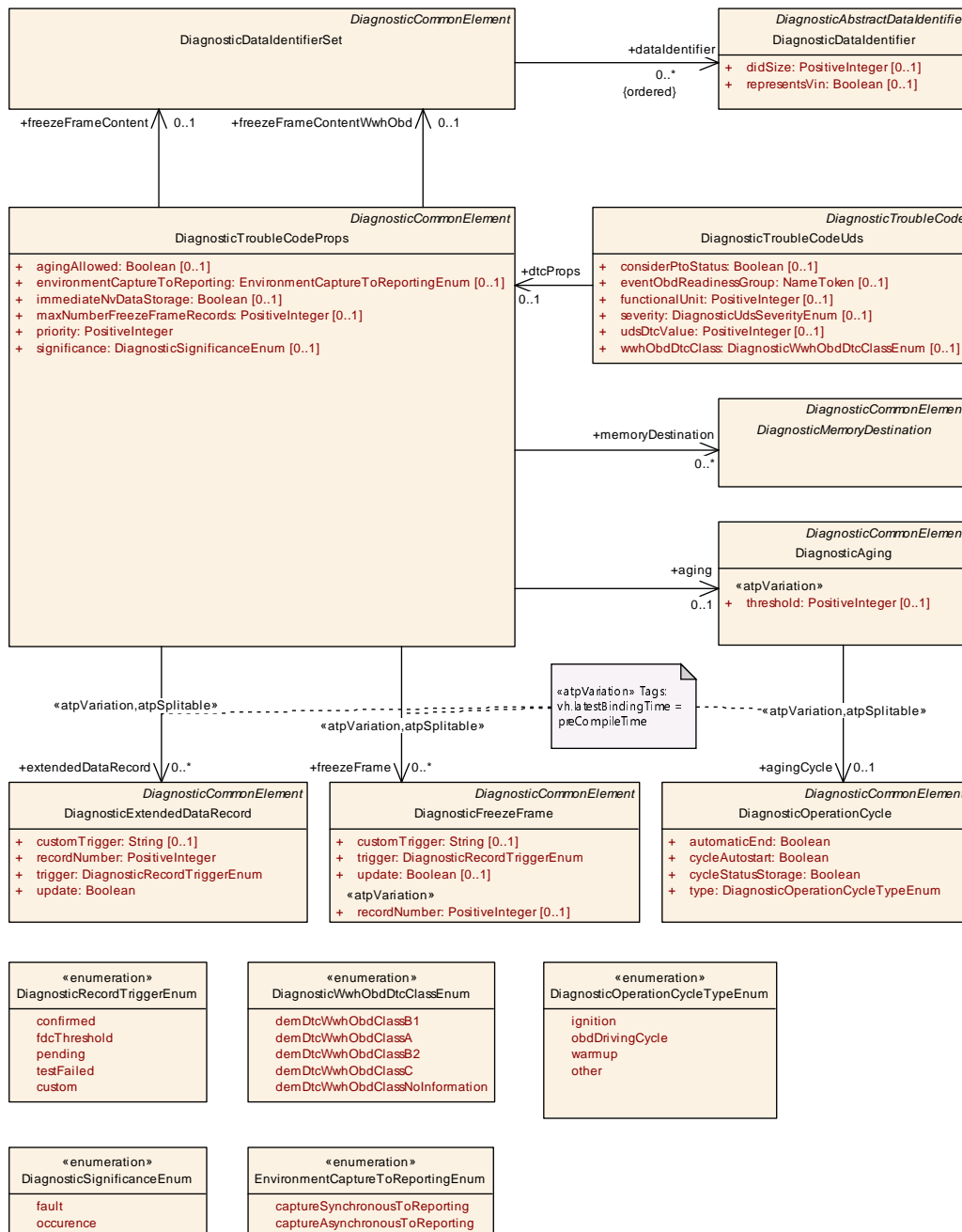


Figure 6.4: Modeling of DiagnosticTroubleCodeUds

[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. ] (RS\_DEXT\_00024)

[constr\_1349] Value of udsDtcValue shall be unique [ The value of udsDtcValue shall be unique to any other DTC and DTC group value. ] ()

<b>Class</b>	<i>DiagnosticTroubleCode</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
<b>Note</b>	A diagnostic trouble code defines a unique identifier that is shown to the diagnostic tester.			
<b>Base</b>	<i>ARElement</i> , <i>ARObject</i> , <i>CollectableElement</i> , <i>DiagnosticCommonElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
<b>Subclasses</b>	<i>DiagnosticTroubleCodeJ1939</i> , <i>DiagnosticTroubleCodeObd</i> , <i>DiagnosticTroubleCodeUds</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 6.11: DiagnosticTroubleCode**

[TPS\_DEXT\_03014] Semantics of **DiagnosticTroubleCodeGroup** [ The *DiagnosticTroubleCodeGroup* element is used to define groups of DTCs that belong together. Each *DiagnosticTroubleCodeGroup* has its own *groupNumber* value assigned. ] (*RS\_DEXT\_00024*)

<b>Class</b>	<b>DiagnosticTroubleCodeGroup</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
<b>Note</b>	The diagnostic trouble code group defines the DTCs belonging together and thereby forming a group. <b>Tags:</b> atp.recommendedPackage=DiagnosticTroubleCodes			
<b>Base</b>	<i>ARElement</i> , <i>ARObject</i> , <i>CollectableElement</i> , <i>DiagnosticCommonElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dtc	<i>DiagnosticTroubleCode</i>	*	ref	This represents the collection of <i>DiagnosticTroubleCodes</i> defined by this <i>DiagnosticTroubleCodeGroup</i> . <b>Stereotypes:</b> atp.Splittable; atp.Variation <b>Tags:</b> atp.Splitkey=dtc, variationPoint.shortLabel vh.latestBindingTime=postBuild
groupNumber	PositiveInteger	1	attr	This represents the base number of the DTC group. <b>Stereotypes:</b> atp.Variation <b>Tags:</b> vh.latestBindingTime=preCompileTime

**Table 6.12: DiagnosticTroubleCodeGroup**

[constr\_1350] Value of **DiagnosticTroubleCodeGroup.groupNumber** shall be unique [ The value of *DiagnosticTroubleCodeGroup.groupNumber* shall be unique to any other DTC and DTC group value. ]()

[constr\_1351] Value of **DiagnosticTroubleCodeGroup.groupNumber** [ To be compliant to ISO, the value of *DiagnosticTroubleCodeGroup.groupNumber* shall be set as defined in ISO 14229-1 [16]. ]()

[TPS\_DEXT\_03000] ISO 14229-1 reserves values of **DiagnosticTroubleCodeGroup.groupNumber** [ Any values other than those mentioned in [constr\_1351] are reserved by ISO 14229-1 [16]. ] (*RS\_DEXT\_00024*)

[constr\_1352] Existence of **maxNumberFreezeFrameRecords** vs. **freezeFrame** [ If the attribute *DiagnosticTroubleCodeProps.maxNumberFreezeFrameRecords* exists than the attribute *DiagnosticTroubleCodeProps.freezeFrame* shall not exist or vice versa. ]()

**[constr\_1353] Applicability of [1352]** [ **[constr\_1352]** shall apply in the identical way (either one or the other attribute shall exist) for all `DiagnosticTroubleCodeProps` within the context of all `DiagnosticContributionSets` of category `DIAGNOSTIC_ECU_EXTRACT` that refer to the same `EcuInstance`. ]()

**[constr\_1354] Existence of attribute `DiagnosticTroubleCodeProps.freezeFrameContent`** [ If one of the attributes `DiagnosticTroubleCodeProps.maxNumberFreezeFrameRecords` or `DiagnosticTroubleCodeProps.freezeFrame` exists then the attribute `DiagnosticTroubleCodeProps.freezeFrameContent` 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`. ]([RS\\_DEXT\\_00024](#), [RS\\_DEXT\\_00045](#))

**[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`:

- 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`.
- Textual description that characterizes the `DiagnosticTroubleCode` with respect to the *ODX long name* shall be provided by means of the attribute `longName`.

]([RS\\_DEXT\\_00024](#), [RS\\_DEXT\\_00045](#))

<b>Class</b>	<b>TraceableText</b>			
<b>Package</b>	M2::MSR::Documentation::BlockElements::RequirementsTracing			
<b>Note</b>	<p>This meta-class represents the ability to denote a traceable text item such as requirements etc.</p> <p>The following approach applies:</p> <ul style="list-style-type: none"> <li>• <b>shortName</b> represents the tag for tracing</li> <li>• <b>longName</b> represents the head line</li> <li>• <b>category</b> represents the kind of the tagged text</li> </ul>			
<b>Base</b>	<i>ARObject</i> , <i>DocumentViewSelectable</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Paginateable</i> , <i>Referrable</i> , <i>Traceable</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>





Class	TraceableText			
text	<a href="#">DocumentationBlock</a>	1	aggr	This represents the text to which the tag applies. <b>Tags:</b> xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=30 xml.typeElement=false xml.typeWrapperElement=false

**Table 6.13: TraceableText**

<b>Class</b>	<i>MultilanguageReferrable</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable			
<b>Note</b>	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.			
<b>Base</b>	<i>ARObject</i> , <i>Referrable</i>			
<b>Subclasses</b>	Caption, DeflItem, DocumentationContext, <i>Identifiable</i> , SdgCaption, <i>TraceReferrable</i> , <i>Traceable</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
longName	MultilanguageLong Name	0..1	aggr	This specifies the long name of the object. Long name is targeted to human readers and acts like a headline.

**Table 6.14: MultilanguageReferrable**

For more details regarding the modeling of the semi-formal text please refer to Figure 4.3.

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](#).

| ([RS\\_DEXT\\_00001](#), [RS\\_DEXT\\_00024](#), [RS\\_DEXT\\_00045](#))

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](#).

**Listing 6.2: Example for the definition of a semi-formal textual elements in the context of a [DiagnosticTroubleCode](#)**

```

<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
          adipiscing elit</L-1>
      </P>
    </TRACE>
  </INTRODUCTION>
  <DTC-PROPS-REF DEST="DIAGNOSTIC-TROUBLE-CODE-PROPS">/AUTOSAR/
    UseCase_230/ExampleDTC_0001_Props</DTC-PROPS-REF>
  <FUNCTIONAL-UNIT>1</FUNCTIONAL-UNIT>
  <SEVERITY>CHECK-AT-NEXT-HALT</SEVERITY>
  <UDS-DTC-VALUE>0x000001</UDS-DTC-VALUE>
</DIAGNOSTIC-TROUBLE-CODE-UDS>

```

**[constr\_1376] Multiplicity of reference [DiagnosticTroubleCodeProps.memoryDestination](#)** [ For every given [DiagnosticTroubleCodeProps](#), the reference in the role [DiagnosticTroubleCodeProps.memoryDestination](#) shall not exceed the upper multiplicity 2. **[constr\_1377]** applies. ]()

**[constr\_1377] Existence of reference [DiagnosticTroubleCodeProps.memoryDestination](#)** [ The reference [DiagnosticTroubleCodeProps.memoryDestination](#) shall **only** have the upper multiplicity 2 if **one (and only one)** of the referenced [DiagnosticTroubleCodeProps.memoryDestination](#) is a [DiagnosticMemoryDestinationMirror](#). ]()

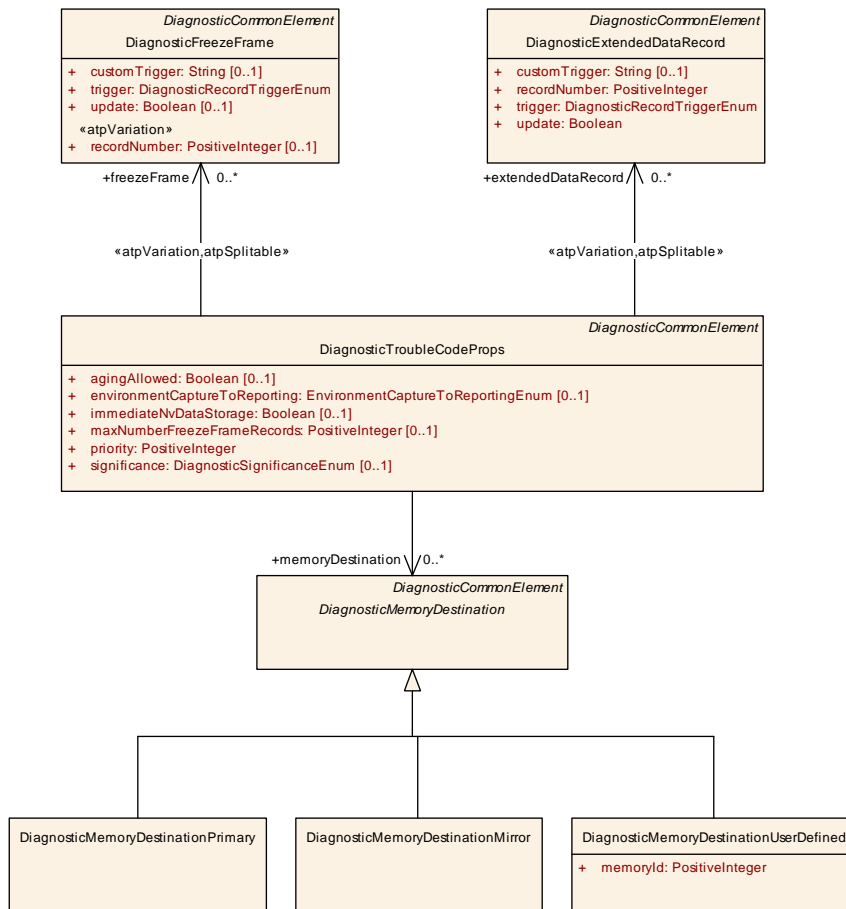


Figure 6.5: Modeling of **DiagnosticMemoryDestination**

**[constr\_1378] Value of [DiagnosticMemoryDestinationUserDefined.memoryId](#)** [ Within the scope of one [DiagnosticContributionSet](#), no two (or more) [DiagnosticMemoryDestinationUserDefineds](#) shall exist that share the same value for attribute [DiagnosticMemoryDestinationUserDefined.memoryId](#) ]()

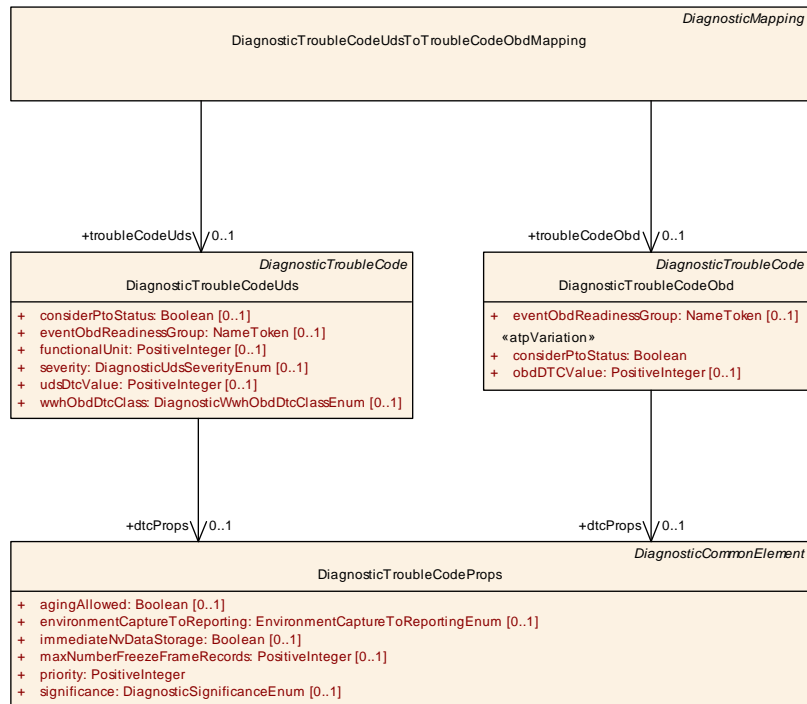
In other words, the value of the attribute [DiagnosticMemoryDestinationUserDefined.memoryId](#) shall be unique within any given [DiagnosticExtract](#).

On top of that, it is necessary to make sure that only **one primary memory** and only **one mirror memory** is defined.

**[constr\_1379] Existence of [DiagnosticMemoryDestinationPrimary](#)** [ Within the scope of one [DiagnosticContributionSet](#) only one [DiagnosticMemoryDestinationPrimary](#) shall exist. ]()

**[constr\_1380] Existence of [DiagnosticMemoryDestinationMirror](#)** [ Within the scope of one [DiagnosticContributionSet](#) only one [DiagnosticMemoryDestinationMirror](#) shall exist. ]()

**[TPS\_DEXT\_01094] Semantics of meta-class [DiagnosticTroubleCodeUdsToTroubleCodeObdMapping](#)** [ The meta-class [DiagnosticTroubleCodeUdsToTroubleCodeObdMapping](#) can be used to associate a [DiagnosticTroubleCodeUds](#) with a [DiagnosticTroubleCodeObd](#). ]()



**Figure 6.6: Mapping of UDS DTC to OBD DTC**

<b>Class</b>	<b>DiagnosticTroubleCodeProps</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
<b>Note</b>	This element defines common Dtc properties that can be reused by different non OBD-relevant DTCs. <b>Tags:</b> atp.recommendedPackage=DiagnosticTroubleCodePropss			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
aging	<a href="#">DiagnosticAging</a>	0..1	ref	Reference to an aging algorithm in case that an aging/unlearning of the event is allowed.
agingAllowed	Boolean	0..1	attr	This represents the decision whether aging is allowed for this DiagnosticTroubleCodeProps.
environmentCaptureToReporting	EnvironmentCaptureToReportingEnum	0..1	attr	This attribute determines the point in time, when the data actually is captured.
extendedDataRecord	<a href="#">DiagnosticExtendedDataRecord</a>	*	ref	Defines the links to an extended data class sampler. <b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
freezeFrame	<a href="#">DiagnosticFreezeFrame</a>	*	ref	Define the links to a freeze frame class sampler. <b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
freezeFrameContent	<a href="#">DiagnosticDataIdentifierSet</a>	0..1	ref	This represents the freeze frame layout as a set of DIDs.
freezeFrameContentWwhObd	<a href="#">DiagnosticDataIdentifierSet</a>	0..1	ref	This reeference identifies the layout of the WWH-OBD freeze frame.







Class	DiagnosticTroubleCodeProps			
immediateNvDataStorage	Boolean	0..1	attr	Switch to enable immediate storage triggering of an according event memory entry persistently to NVRAM. true: immediate non-volatile storage triggering enabled false: immediate non-volatile storage triggering disabled
maxNumberFreezeFrameRecords	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.
memoryDestination	<a href="#">DiagnosticMemoryDestination</a>	*	ref	The event destination assigns events to none, one or multiple origins.
priority	PositiveInteger	1	attr	Priority of the event, in view of full event buffer. A lower value means higher priority.
significance	<a href="#">DiagnosticSignificanceEnum</a>	0..1	attr	Significance of the event, which indicates additional information concerning fault classification and resolution.

**Table 6.15: DiagnosticTroubleCodeProps**

<b>Class</b>	<i>DiagnosticMemoryDestination</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
<b>Note</b>	This abstract meta-class represents a possible memory destination for a diagnostic event.			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">DiagnosticMemoryDestinationMirror</a> , <a href="#">DiagnosticMemoryDestinationPrimary</a> , <a href="#">DiagnosticMemoryDestinationUserDefined</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 6.16: DiagnosticMemoryDestination**

<b>Class</b>	<i>DiagnosticMemoryDestinationPrimary</i>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
<b>Note</b>	This represents a primary memory for a diagnostic event. <b>Tags:</b> atp.recommendedPackage=DiagnosticMemoryDestinations			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMemoryDestination</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 6.17: DiagnosticMemoryDestinationPrimary**

<b>Class</b>	<i>DiagnosticMemoryDestinationMirror</i>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
<b>Note</b>	This represents a mirror memory for a diagnostic event. <b>Tags:</b> atp.recommendedPackage=DiagnosticMemoryDestinations			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMemoryDestination</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–





<b>Class</b>	<b>DiagnosticMemoryDestinationMirror</b>			
-	-	-	-	-

**Table 6.18: DiagnosticMemoryDestinationMirror**

<b>Class</b>	<b>DiagnosticMemoryDestinationUserDefined</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
<b>Note</b>	This represents a user-defined memory for a diagnostic event. <b>Tags:</b> atp.recommendedPackage=DiagnosticMemoryDestinations			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMemoryDestination</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
memoryId	PositiveInteger	1	attr	This represents the identifier of the user-defined memory.

**Table 6.19: DiagnosticMemoryDestinationUserDefined**

<b>Enumeration</b>	<b>DiagnosticSignificanceEnum</b>		
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode		
<b>Note</b>	Significance level of a diagnostic event.		
<b>Literal</b>	<b>Description</b>		
fault	Failure, which affects the component/ECU itself. <b>Tags:</b> atp.EnumerationValue=0		
occurrence	Issue, which indicates additional information concerning insufficient system behavior. <b>Tags:</b> atp.EnumerationValue=1		

**Table 6.20: DiagnosticSignificanceEnum**

<b>Enumeration</b>	<b>DiagnosticUdsSeverityEnum</b>		
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode		
<b>Note</b>	Severity types for a DTC according to ISO 14229-1.		
<b>Literal</b>	<b>Description</b>		
checkAtNextHalt	Check at next halt. <b>Tags:</b> atp.EnumerationValue=0		
immediately	Check immediately. <b>Tags:</b> atp.EnumerationValue=1		
maintenanceOnly	Maintenance required. <b>Tags:</b> atp.EnumerationValue=2		
noSeverity	No severity information available. <b>Tags:</b> atp.EnumerationValue=3		

**Table 6.21: DiagnosticUdsSeverityEnum**

<b>Class</b>	<b>DiagnosticDataIdentifierSet</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
<b>Note</b>	This represents the ability to define a list of DiagnosticDataIdentifiers that can be reused in different contexts. <b>Tags:</b> atp.recommendedPackage=DiagnosticDataIdentifierSets			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataIdentifier (ordered)	<a href="#">DiagnosticDataIdentifier</a>	*	ref	Reference to an ordered list of Data Identifiers.

**Table 6.22: DiagnosticDataIdentifierSet**

<b>Enumeration</b>	<b>DiagnosticWwhObdDtcClassEnum</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
<b>Note</b>	This meta-class represents the ability to model severity classes of an WWH-OB DTC.			
<b>Literal</b>	<b>Description</b>			
demDtcWwhObdClassA	This attribute represents the severity class A. <b>Tags:</b> atp.EnumerationValue=0			
demDtcWwhObdClassB1	This attribute represents the severity class B1. <b>Tags:</b> atp.EnumerationValue=1			
demDtcWwhObdClassB2	This attribute represents the severity class B2. <b>Tags:</b> atp.EnumerationValue=2			
demDtcWwhObdClassC	This attribute represents the severity class C. <b>Tags:</b> atp.EnumerationValue=3			
demDtcWwhObdClassNoInformation	This attribute represents the option to intentionally not describe a dedicated severity class of an WWH-OB DTC. <b>Tags:</b> atp.EnumerationValue=4			

**Table 6.23: DiagnosticWwhObdDtcClassEnum**

<b>Class</b>	<b>DiagnosticTroubleCodeUdsToTroubleCodeObdMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
<b>Note</b>	This meta-class represents the ability to associate a UDS trouble code to an OBD trouble code. <b>Tags:</b> atp.recommendedPackage=DiagnosticMappings			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMapping</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
troubleCodeObd	<a href="#">DiagnosticTroubleCodeObd</a>	0..1	ref	This represents the OBD DTC referenced in the mapping between UDS and OBD DTCs.
troubleCodeUds	<a href="#">DiagnosticTroubleCodeUds</a>	0..1	ref	This represents the UDS DTC referenced in the mapping between UDS and OBD DTCs.

**Table 6.24: DiagnosticTroubleCodeUdsToTroubleCodeObdMapping**

## 6.4 DiagnosticExtendedDataRecord

[TPS\_DEXT\_03008] Semantics of **DiagnosticExtendedDataRecord** [ A `DiagnosticExtendedDataRecord` contains `DiagnosticDataElements` that are ordered by the `bitOffset`. ](*RS\_DEXT\_00032*)

[constr\_1355] Value of **extendedDataRecord.recordNumber** [ To be compliant to ISO, the value of `extendedDataRecord.recordNumber` shall be set in the interval as defined in ISO 14229-1 [16]. ]()

[constr\_1509] **extendedDataRecord.recordNumber** shall be unique within primary fault memory [ For all `DiagnosticTroubleCodeProps` that refer to `DiagnosticMemoryDestinationPrimary` in the role `memoryDestination` there shall be no two `extendedDataRecord.recordNumber` with the same value. ]()

[constr\_1510] **extendedDataRecord.recordNumber** shall be unique within mirror fault memory [ For all `DiagnosticTroubleCodeProps` that refer to `DiagnosticMemoryDestinationMirror` in the role `memoryDestination` there shall be no two `extendedDataRecord.recordNumber` with the same value. ]()

[constr\_1511] **extendedDataRecord.recordNumber** shall be unique within user-defined fault memory [ For all `DiagnosticTroubleCodeProps` that refer to `DiagnosticMemoryDestinationUserDefined` in the role `memoryDestination` there shall be no two `extendedDataRecord.recordNumber` with the same value for any `DiagnosticMemoryDestinationUserDefined` referenced as `DiagnosticTroubleCodeProps.memoryDestination` with a given value of `memoryId`. ]()

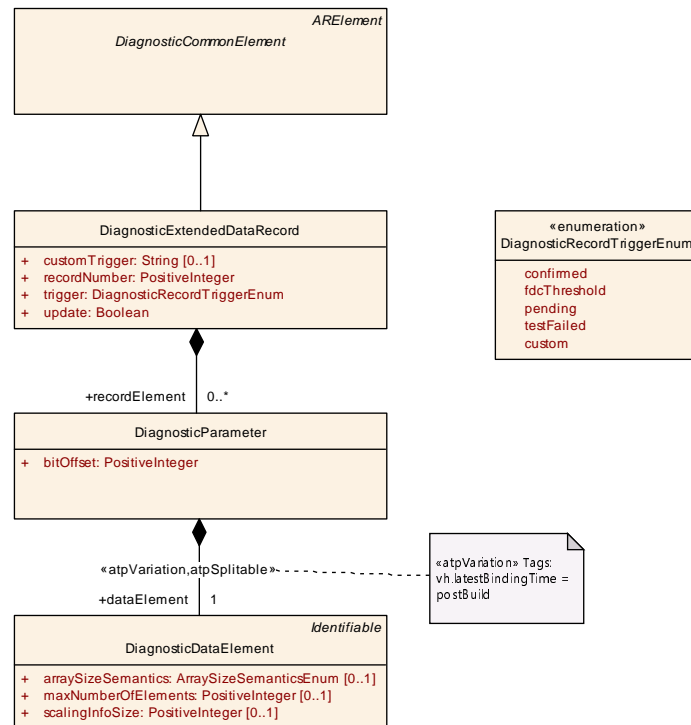


Figure 6.7: Modeling of DiagnosticExtendedDataRecord

**[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. ]([RS\\_DEXT\\_00033](#))

**[constr\_1616] Existence of attribute `DiagnosticExtendedDataRecord.customTrigger`** [ The attribute `DiagnosticExtendedDataRecord.customTrigger` shall only exist if the attribute `DiagnosticExtendedDataRecord.trigger` is set to the value `DiagnosticRecordTriggerEnum.custom`. ]()

<b>Class</b>	<b>DiagnosticExtendedDataRecord</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticExtendedDataRecord			
<b>Note</b>	Description of an extended data record. <b>Tags:</b> atp.recommendedPackage=DiagnosticExtendedDataRecords			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
customTrigger	String	0..1	attr	This attribute shall be taken to verbally describe the nature of the custom trigger.
recordElement	<a href="#">DiagnosticParameter</a>	*	aggr	Defined DataElements in the extended record element.
recordNumber	PositiveInteger	1	attr	This attribute specifies a unique identifier for an extended data record.
trigger	<a href="#">DiagnosticRecordTriggerEnum</a>	1	attr	This attribute specifies the primary trigger to allocate an event memory entry.
update	Boolean	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 6.25: DiagnosticExtendedDataRecord**

<b>Enumeration</b>	<b>DiagnosticRecordTriggerEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticFreezeFrame
<b>Note</b>	Triggers to allocate an event memory entry.
<b>Literal</b>	<b>Description</b>
confirmed	capture on "Confirmed" <b>Tags:</b> atp.EnumerationValue=0
custom	implement custom capture <b>Tags:</b> atp.EnumerationValue=4
fdcThreshold	capture on "FDC Threshold" <b>Tags:</b> atp.EnumerationValue=1
pending	capture on "Pending" <b>Tags:</b> atp.EnumerationValue=2





Enumeration	DiagnosticRecordTriggerEnum
testFailed	capture on "Test Failed" Tags: atp.EnumerationValue=3

**Table 6.26: DiagnosticRecordTriggerEnum**

## 6.5 DiagnosticFreezeFrame

[TPS\_DEXT\_03009] **Semantics of DiagnosticFreezeFrame** [ A DiagnosticFreezeFrame needs an ordered list of references to DiagnosticDataIdentifiers. However, this reference is not modeled directly but in the context of meta-class DiagnosticTroubleCodeProps. ](RS\_DEXT\_00033)

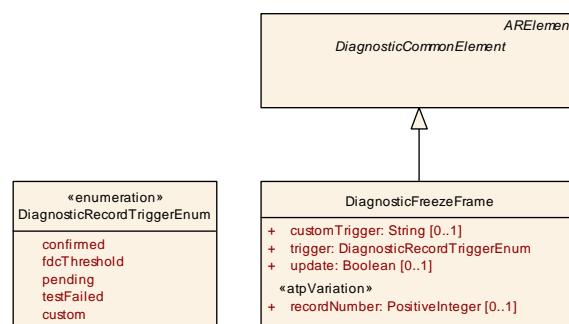
For more details, please refer to Figure 6.3.

[constr\_1357] **Value of freezeFrame.recordNumber** [ To be compliant to ISO, the value of freezeFrame.recordNumber shall be set in the interval as defined in ISO 14229-1 [16]. ]()

[constr\_1512] **freezeFrame.recordNumber shall be unique within primary fault memory** [ For all DiagnosticTroubleCodeProps that refer to DiagnosticMemoryDestinationPrimary in the role memoryDestination there shall be no two freezeFrame.recordNumber with the same value. ]()

[constr\_1513] **freezeFrame.recordNumber shall be unique within mirror fault memory** [ For all DiagnosticTroubleCodeProps that refer to DiagnosticMemoryDestinationMirror in the role memoryDestination there shall be no two freezeFrame.recordNumber with the same value. ]()

[constr\_1514] **freezeFrame.recordNumber shall be unique within user-defined fault memory** [ For all DiagnosticTroubleCodeProps that refer to DiagnosticMemoryDestinationUserDefined in the role memoryDestination there shall be no two freezeFrame.recordNumber with the same value for any DiagnosticMemoryDestinationUserDefined referenced as DiagnosticTroubleCodeProps.memoryDestination with a given value of memoryId. ]()



**Figure 6.8: Modeling of 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. ]([RS\\_DEXT\\_00033](#))

**[constr\_1617] Existence of attribute `DiagnosticFreezeFrame.customTrigger`** [ The attribute `DiagnosticFreezeFrame.customTrigger` shall only exist if the attribute `DiagnosticFreezeFrame.trigger` is set to the value `DiagnosticRecordTriggerEnum.custom`. ]()

<b>Class</b>	<b>DiagnosticFreezeFrame</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticFreezeFrame			
<b>Note</b>	This element describes combinations of DIDs for a non OBD relevant freeze frame. <b>Tags:</b> atp.recommendedPackage=DiagnosticFreezeFrames			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
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. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime
trigger	<a href="#">DiagnosticRecordTriggerEnum</a>	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 6.27: DiagnosticFreezeFrame**

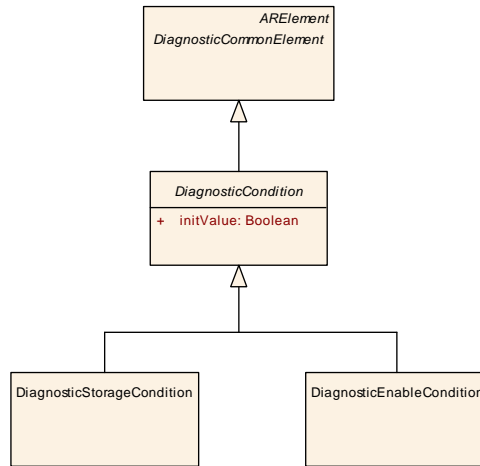
## 6.6 DiagnosticCondition

**[TPS\_DEXT\_03010] Combination of `DiagnosticConditions` to `DiagnosticConditionGroups`** [ `DiagnosticConditions` are combined to `DiagnosticConditionGroups` and define a certain number of checks (e.g. correct voltage range) before the event report is accepted or the event gets qualified. ]([RS\\_DEXT\\_00027](#), [RS\\_DEXT\\_00028](#), [RS\\_DEXT\\_00030](#), [RS\\_DEXT\\_00031](#))

**[TPS\_DEXT\_03001] Different types of conditions** [ There are two different types of conditions: `DiagnosticEnableConditions` and `DiagnosticStorageCondition`:

- 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.

|(RS\_DEXT\_00027)



**Figure 6.9: Modeling of DiagnosticCondition**

<b>Class</b>	<i>DiagnosticCondition</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticCondition			
<b>Note</b>	Abstract element for StorageConditions and EnableConditions.			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">DiagnosticEnableCondition</a> , <a href="#">DiagnosticStorageCondition</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
initValue	Boolean	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 6.28: DiagnosticCondition**

<b>Class</b>	<b>DiagnosticEnableCondition</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticCondition			
<b>Note</b>	Specification of an enable condition. <b>Tags:</b> atp.recommendedPackage=DiagnosticConditions			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticCondition</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 6.29: DiagnosticEnableCondition**

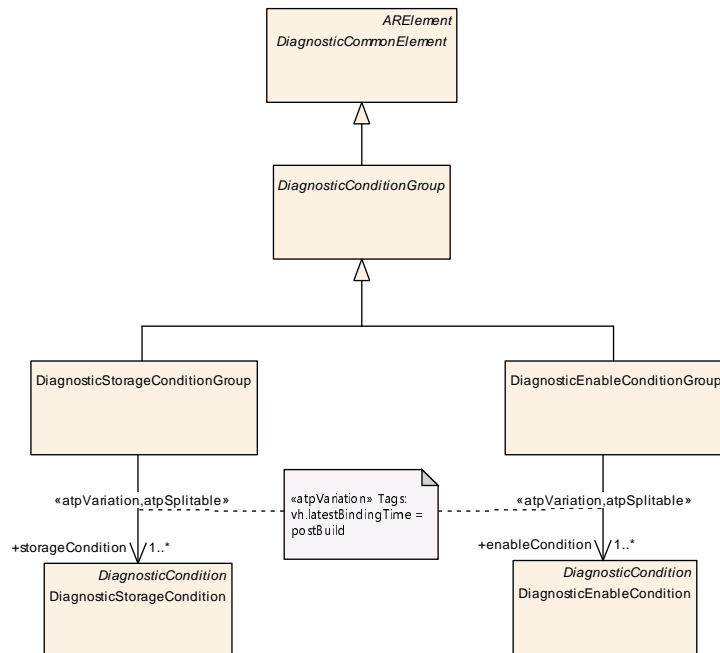


<b>Class</b>	<b>DiagnosticStorageCondition</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticCondition			
<b>Note</b>	Specification of a storage condition. <b>Tags:</b> atp.recommendedPackage=DiagnosticConditions			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticCondition</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
-	-	-	-	-

**Table 6.30: DiagnosticStorageCondition**

## 6.7 DiagnosticConditionGroup

[TPS\_DEXT\_01084] Semantics of **DiagnosticConditionGroups** [ [DiagnosticConditionGroups](#) are used to collect [DiagnosticConditions](#) that in turn are assigned to [DiagnosticEvents](#). ] ([RS\\_DEXT\\_00023](#), [RS\\_DEXT\\_00028](#), [RS\\_DEXT\\_00029](#))



**Figure 6.10: Modeling of DiagnosticConditionGroup**

<b>Class</b>	<b>DiagnosticConditionGroup</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticConditionGroup			
<b>Note</b>	Abstract element for StorageConditionGroups and EnableConditionGroups.			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			





<b>Class</b>	<i>DiagnosticConditionGroup</i> (abstract)			
<b>Subclasses</b>	<a href="#">DiagnosticEnableConditionGroup</a> , <a href="#">DiagnosticStorageConditionGroup</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 6.31: DiagnosticConditionGroup**

<b>Class</b>	<b>DiagnosticEnableConditionGroup</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticConditionGroup			
<b>Note</b>	Enable condition group which includes one or several enable conditions. <b>Tags:</b> atp.recommendedPackage=DiagnosticConditions			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticConditionGroup</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
enableCondition	<a href="#">DiagnosticEnableCondition</a>	1..*	ref	Reference to enableConditions that are part of the EnableConditionGroup. <b>Stereotypes:</b> atp.Splittable; atp.Variation <b>Tags:</b> atp.Splitkey=enableCondition, variationPoint.shortLabel vh.latestBindingTime=postBuild

**Table 6.32: DiagnosticEnableConditionGroup**

<b>Class</b>	<b>DiagnosticStorageConditionGroup</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticConditionGroup			
<b>Note</b>	Storage condition group which includes one or several storage conditions. <b>Tags:</b> atp.recommendedPackage=DiagnosticConditions			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticConditionGroup</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
storageCondition	<a href="#">DiagnosticStorageCondition</a>	1..*	ref	Reference to storageConditions that are part of the StorageConditionGroup. <b>Stereotypes:</b> atp.Splittable; atp.Variation <b>Tags:</b> atp.Splitkey=storageCondition, variationPoint.shortLabel vh.latestBindingTime=postBuild

**Table 6.33: DiagnosticStorageConditionGroup**

## 6.8 DiagnosticMapping

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.

[TPS\_DEXT\_03002] Two kind of mappings [ For diagnostic event handling, there are two kind of mappings:

- Mapping between a *DiagnosticEvent* and another diagnostic definition.
- Mapping between a *DiagnosticEvent* and a SWC service port.

]([RS\\_DEXT\\_00023](#), [RS\\_DEXT\\_00052](#))

Figure 6.11 gives an overview on the different types of mappings available for diagnostic event handling.

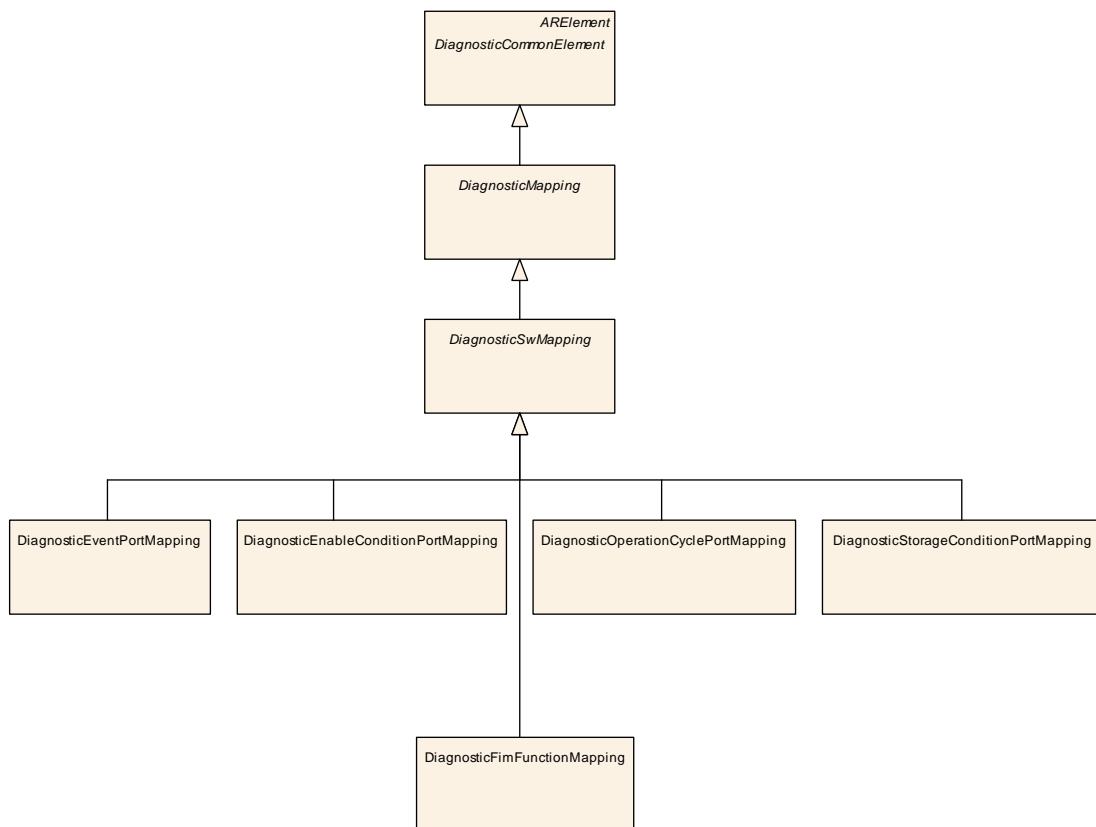


Figure 6.11: Modeling of DiagnosticMapping

<b>Class</b>	<i>DiagnosticMapping</i> (abstract)
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping
<b>Note</b>	Abstract element for different kinds of diagnostic mappings.
<b>Base</b>	<i>ARElement</i> , <i>ARObject</i> , <i>CollectableElement</i> , <i>DiagnosticCommonElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>
<b>Subclasses</b>	<i>DiagnosticDemProvidedDataMapping</i> , <i>DiagnosticEventToDebounceAlgorithmMapping</i> , <i>DiagnosticEventToEnableConditionGroupMapping</i> , <i>DiagnosticEventToOperationCycleMapping</i> , <i>DiagnosticEventToStorageConditionGroupMapping</i> , <i>DiagnosticEventToTroubleCodeJ1939Mapping</i> , <i>DiagnosticEventToTroubleCodeUdsMapping</i> , <i>DiagnosticFimAliasEventGroupMapping</i> , <i>DiagnosticFimAliasEventMapping</i> , <i>DiagnosticInhibitSourceEventMapping</i> , <i>DiagnosticJ1939SpnMapping</i> , <i>DiagnosticServiceDataMapping</i> , <i>DiagnosticSwMapping</i> , <i>DiagnosticTroubleCodeUdsToTroubleCodeObdMapping</i>





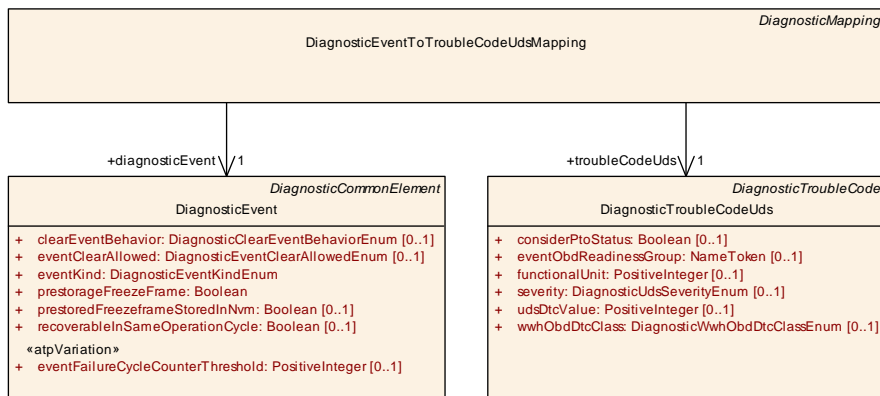
Class	DiagnosticMapping (abstract)			
Attribute	Type	Mul.	Kind	Note
–	–	–	–	–

**Table 6.34: DiagnosticMapping**

### 6.8.1 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 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. ] ([RS\\_DEXT\\_00023](#), [RS\\_DEXT\\_00024](#), [RS\\_DEXT\\_00025](#))



**Figure 6.12: DiagnosticEventToDtcUdsMapping**

Class	DiagnosticEventToTroubleCodeUdsMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
Note	Defines which UDS Diagnostic Trouble Code is applicable for a DiagnosticEvent. Tags: atp.recommendedPackage=DiagnosticMappings			
Base	<a href="#">ARElement</a> , <a href="#">AObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMapping</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
Attribute	Type	Mul.	Kind	Note
diagnosticEvent	<a href="#">DiagnosticEvent</a>	1	ref	Reference to a DiagnosticEvent to which a UDS Diagnostic Trouble Code is assigned.
troubleCodeUds	<a href="#">DiagnosticTroubleCodeUds</a>	1	ref	Reference to an UDS Diagnostic Trouble Code assigned to a DiagnosticEvent.

**Table 6.35: DiagnosticEventToTroubleCodeUdsMapping**

### 6.8.2 DiagnosticEvent to DiagnosticOperationCycle Mapping

[TPS\_DEXT\_01086] Reference to [DiagnosticOperationCycle](#) [ A [DiagnosticEvent](#) needs to be assigned to exactly one [DiagnosticOperationCycle](#). ] (RS\_DEXT\_00024, RS\_DEXT\_00054)

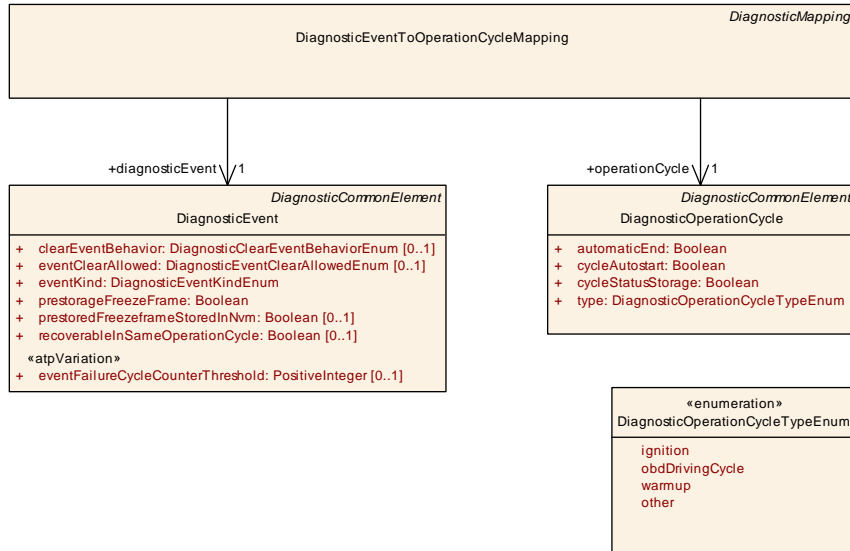


Figure 6.13: DiagnosticEventToOperationCycleMapping

<b>Class</b>	<b>DiagnosticEventToOperationCycleMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
<b>Note</b>	Defines which OperationCycle is applicable for a DiagnosticEvent. <b>Tags:</b> atp.recommendedPackage=DiagnosticMappings			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMapping</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
diagnosticEvent	<a href="#">DiagnosticEvent</a>	1	ref	Reference to a DiagnosticEvent to which an Operation Cycle is assigned.
operationCycle	<a href="#">DiagnosticOperation Cycle</a>	1	ref	Reference to an OperationCycle assigned to a Diagnostic Event.

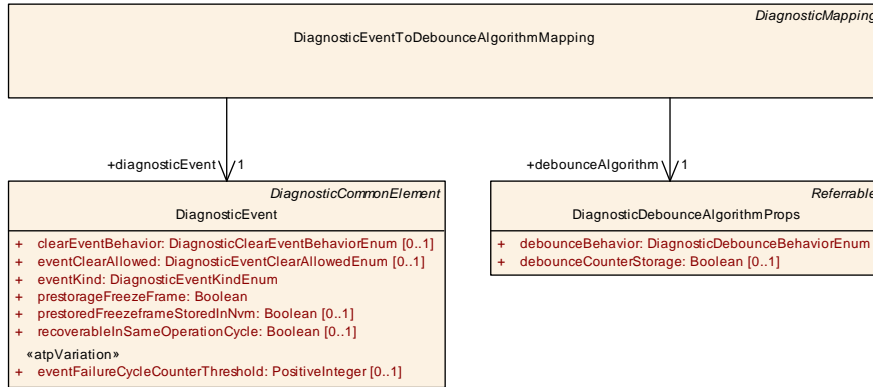
Table 6.36: DiagnosticEventToOperationCycleMapping

### 6.8.3 DiagnosticEvent to DebounceAlgorithm Mapping

[TPS\_DEXT\_03004] [DiagnosticEvent](#) and [DiagnosticDebounceAlgorithm-Props](#) [ If a [DiagnosticEvent](#) has to be debounced, it must be mapped to the appropriate [DiagnosticDebounceAlgorithmProps](#). ] (RS\_DEXT\_00023, RS\_DEXT\_00053)

[TPS\_DEXT\_03005] Existence of [DiagnosticEventToDebounceAlgorithmMapping](#) [ The [DiagnosticEventToDebounceAlgorithmMapping](#) shall not

be created if the `DiagnosticEvent` is not debounced. [|\(RS\\_DEXT\\_00023, RS\\_DEXT\\_00053\)](#)



**Figure 6.14: DiagnosticEventToDebounceAlgorithmMapping**

<b>Class</b>	<b>DiagnosticEventToDebounceAlgorithmMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
<b>Note</b>	Defines which Debounce Algorithm is applicable for a DiagnosticEvent. <b>Tags:</b> atp.recommendedPackage=DiagnosticMappings			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMapping</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
debounce Algorithm	<a href="#">DiagnosticDebounceAlgorithmProps</a>	1	ref	Reference to a DebounceAlgorithm assigned to a DiagnosticEvent.
diagnosticEvent	<a href="#">DiagnosticEvent</a>	1	ref	Reference to a DiagnosticEvent to which a Debounce Algorithm is assigned.

**Table 6.37: DiagnosticEventToDebounceAlgorithmMapping**

<b>Class</b>	<b>DiagnosticDebounceAlgorithmProps</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticDebouncingAlgorithm			
<b>Note</b>	Defines properties for the debounce algorithm class.			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
debounce Algorithm	<a href="#">DiagEventDebounceAlgorithm</a>	1	aggr	This represents the actual debounce algorithm.
debounce Behavior	<a href="#">DiagnosticDebounceBehaviorEnum</a>	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.
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

**Table 6.38: DiagnosticDebounceAlgorithmProps**

The details regarding the formalization of debouncing behavior are depicted in [Figure 6.15](#).

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.

**[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`. ]([RS\\_DEXT\\_00023](#), [RS\\_DEXT\\_00053](#))

In other words, the debouncing of diagnostic events can be formulated in two ways:

- The `DiagEventDebounceCounterBased` represents the ability to implement a counter-based debouncing.
- The `DiagEventDebounceTimeBased` represents the ability to implement a time-based debouncing.

**[constr\_1359] Existence of attribute `DiagnosticDebounceAlgorithmProps.debounceCounterStorage`** [ The attribute `DiagnosticDebounceAlgorithmProps.debounceCounterStorage` shall only exist if the aggregation `DiagnosticDebounceAlgorithmProps.debounceAlgorithm` actually aggregates a `DiagEventDebounceCounterBased` ]()

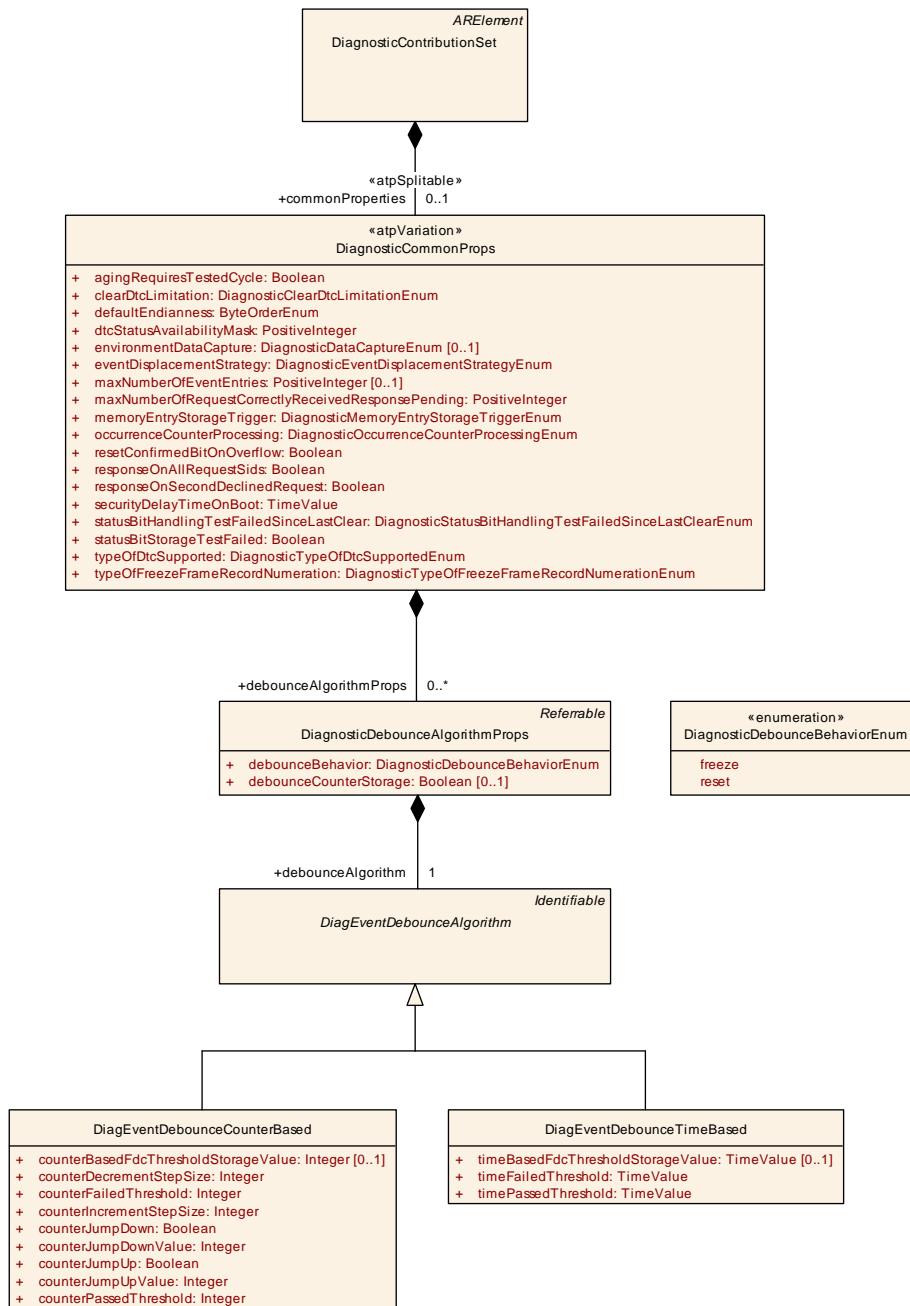


Figure 6.15: Details of **DiagnosticDebounceAlgorithmProps**

[constr\_1360] Usage of **DiagEventDebounceMonitorInternal** is not supported in the context of **DiagnosticDebounceAlgorithmProps** [ The usage of the meta-class **DiagEventDebounceMonitorInternal** for the aggregation in the role **DiagnosticDebounceAlgorithmProps.debounceAlgorithm** is not permitted. ]()

For clarification with respect to [constr\_1360], **DiagEventDebounceMonitorInternal** is used in the context of formulating the **DiagnosticEventNeeds**, but its usage in the context of the **DiagnosticExtract** is not foreseen.



<b>Class</b>				
<b>DiagnosticEventNeeds</b>				
<b>Package</b>				
M2::AUTOSARTemplates::CommonStructure::ServiceNeeds				
<b>Note</b>				
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.				
<b>Base</b>				
ARObject, DiagnosticCapabilityElement, Identifiable, MultilanguageReferrable, Referrable, ServiceNeeds				
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
considerPtoStatus	Boolean	0..1	attr	PTO (Power Take Off) has an impact on the respective emission-related event (OBD). This information shall be provided by SW-C description in order to consider the PTO relevance e.g. for readiness (PID \$01) computation. For events with dtcKind set to 'nonEmissionRelatedDtc' this attribute is typically false.
deferringFid	<a href="#">FunctionInhibitionNeeds</a>	*	ref	This reference contains the link to a function identifier within the FiM which is used by the monitor before delivering a result.
diagEventDebounceAlgorithm	<a href="#">DiagEventDebounceAlgorithm</a>	0..1	aggr	Specifies the abstract need on the Debounce Algorithm applied by the Diagnostic Event Manager.
dtcKind	DtcKindEnum	0..1	attr	This attribute indicates the kind of the diagnostic monitor according to the SWS Diagnostic Event Manger.  This attribute applies for the UDS diagnostics use case.
inhibitingFid	<a href="#">FunctionInhibitionNeeds</a>	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.
inhibitingSecondaryFid	<a href="#">FunctionInhibitionNeeds</a>	*	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.
obdDtcNumber	PositiveInteger	0..1	attr	This represents a reasonable Diagnostic Trouble Code. This allows to predefine the Diagnostic Trouble Code, e.g. if the a function developer has received a particular requirement from the OEM or from a standardization body.  This attribute applies for the OBD diagnostics use case.
prestoredFreezeFrameStoredInNvm	Boolean	0..1	attr	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).
reportBehavior	ReportBehaviorEnum	0..1	attr	This switch indicates whether or not the BSW module is allowed to report the related Events before Dem_Init().
udsDtcNumber	PositiveInteger	0..1	attr	This represents a reasonable Diagnostic Trouble Code. This allows to predefine the Diagnostic Trouble Code, e.g. if the a function developer has received a particular requirement from the OEM or from a standardization body.  This attribute applies for the UDS diagnostics use case.
usesMonitorData	Boolean	0..1	attr	This attribute defines whether additional monitor data shall be added to the reporting of events.

**Table 6.39: DiagnosticEventNeeds**

<b>Class</b>	<i>DiagEventDebounceAlgorithm</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
<b>Note</b>	<p>This class represents the ability to specify the pre-debounce algorithm which is selected and/or required by the particular monitor.</p> <p>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.</p>			
<b>Base</b>	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
<b>Subclasses</b>	<i>DiagEventDebounceCounterBased</i> , <i>DiagEventDebounceMonitorInternal</i> , <i>DiagEventDebounceTimeBased</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 6.40: DiagEventDebounceAlgorithm**

<b>Class</b>	<b>DiagEventDebounceCounterBased</b>			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
<b>Note</b>	<p>This meta-class represents the ability to indicate that the counter-based debounce algorithm shall be used by the DEM for this diagnostic monitor.</p> <p>This is related to set the ECUC choice container DemDebounceAlgorithmClass to DemDebounceCounterBased.</p>			
<b>Base</b>	<i>ARObject</i> , <i>DiagEventDebounceAlgorithm</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
counterBasedFdcThresholdStorageValue	Integer	0..1	attr	Threshold to allocate an event memory entry and to capture the Freeze Frame.
counterDecrementStepSize	Integer	1	attr	This value shall be taken to decrement the internal debounce counter.
counterFailedThreshold	Integer	1	attr	This value defines the event-specific limit that indicates the "failed" counter status.
counterIncrementStepSize	Integer	1	attr	This value shall be taken to increment the internal debounce counter.
counterJumpDown	Boolean	1	attr	This value activates or deactivates the counter jump-down behavior.
counterJumpDownValue	Integer	1	attr	This value represents the initial value of the internal debounce counter if the counting direction changes from incrementing to decrementing.
counterJumpUp	Boolean	1	attr	This value activates or deactivates the counter jump-up behavior.
counterJumpUpValue	Integer	1	attr	This value represents the initial value of the internal debounce counter if the counting direction changes from decrementing to incrementing.
counterPassedThreshold	Integer	1	attr	This value defines the event-specific limit that indicates the "passed" counter status.

**Table 6.41: DiagEventDebounceCounterBased**

<b>Class</b>	<b>DiagEventDebounceTimeBased</b>			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
<b>Note</b>	<p>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.</p> <p>This is related to set the EcuC choice container DemDebounceAlgorithmClass to DemDebounceTimeBase.</p>			
<b>Base</b>	ARObject, <a href="#">DiagEventDebounceAlgorithm</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
timeBasedFdcThresholdStorageValue	TimeValue	0..1	attr	Threshold to allocate an event memory entry and to capture the Freeze Frame.
timeFailedThreshold	TimeValue	1	attr	This value represents the event-specific delay indicating the "failed" status.
timePassedThreshold	TimeValue	1	attr	This value represents the event-specific delay indicating the "passed" status.

**Table 6.42: DiagEventDebounceTimeBased**

<b>Enumeration</b>	<b>DiagnosticDebounceBehaviorEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticDebouncingAlgorithm
<b>Note</b>	Event debounce algorithm behavior options.
<b>Literal</b>	<b>Description</b>
freeze	<p>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).</p> <p><b>Tags:</b> atp.EnumerationValue=0</p>
reset	<p>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.</p> <p><b>Tags:</b> atp.EnumerationValue=1</p>

**Table 6.43: DiagnosticDebounceBehaviorEnum**

<b>Class</b>	<b>DiagEventDebounceMonitorInternal</b>			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
<b>Note</b>	<p>This meta-class represents the ability to indicate that the pre-debounce algorithm shall be used by the Dem for this diagnostic monitor.</p> <p>This is related to setting the EcuC choice container DemDebounceAlgorithmClass to DemDebounceMonitorInternal.</p> <p>If the FaultDetectionAlgorithm is already known to be implemented by a specific BswModuleEntry the reference bswModuleEntry points to the function specification.</p> <p>If the FaultDetectionCounter value is accessible at a PortPrototype this PortPrototype shall be referenced by an assignedPort.</p>			
<b>Base</b>	ARObject, <a href="#">DiagEventDebounceAlgorithm</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 6.44: DiagEventDebounceMonitorInternal**

### 6.8.4 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. ] (RS\_DEXT\_00023, RS\_DEXT\_00026, RS\_DEXT\_00028)

[constr\_1361] **Number of DiagnosticEventToEnableConditionGroupMapping elements per DiagnosticEvent** [ The mapping element DiagnosticEventToEnableConditionGroupMapping shall be created no more than once per DiagnosticEvent.

If several DiagnosticEventToEnableConditionGroupMapping elements referring the same DiagnosticEvent are defined, then the Enable Condition Group mapping shall be regarded as defective. ]()

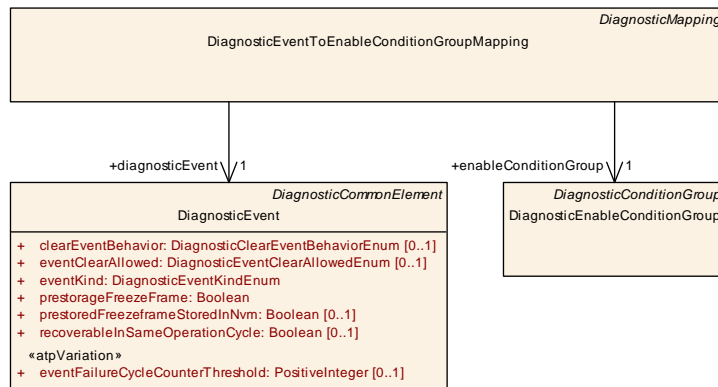


Figure 6.16: DiagnosticEventToEnableConditionGroupMapping

<b>Class</b>	<b>DiagnosticEventToEnableConditionGroupMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
<b>Note</b>	Defines which EnableConditionGroup is applicable for a DiagnosticEvent. <b>Tags:</b> atp.recommendedPackage=DiagnosticMappings			
<b>Base</b>	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
diagnosticEvent	DiagnosticEvent	1	ref	Reference to a DiagnosticEvent to which an EnableConditionGroup is assigned.
enableConditionGroup	DiagnosticEnableConditionGroup	1	ref	Reference to an EnableConditionGroup assigned to a DiagnosticEvent.

Table 6.45: DiagnosticEventToEnableConditionGroupMapping

### 6.8.5 DiagnosticEvent to StorageConditionGroup Mapping

[TPS\_DEXT\_03016] **StorageConditions have to be put into a DiagnosticStorageConditionGroup** [ StorageConditions that are assigned to a DiagnosticEvent have to be put into a DiagnosticStorageConditionGroup since only a group of StorageConditions can be mapped to a DiagnosticEvent. ] (RS\_DEXT\_00023, RS\_DEXT\_00027, RS\_DEXT\_00029)

[constr\_1362] **Number of DiagnosticEventToStorageConditionGroupMapping elements per DiagnosticEvent** [ The mapping element DiagnosticEventToStorageConditionGroupMapping shall be created no more than once or once per DiagnosticEvent.

If several DiagnosticEventToStorageConditionGroupMapping elements referring the same DiagnosticEvent are defined, then the Storage Condition Group mapping shall be regarded as defective. ]()

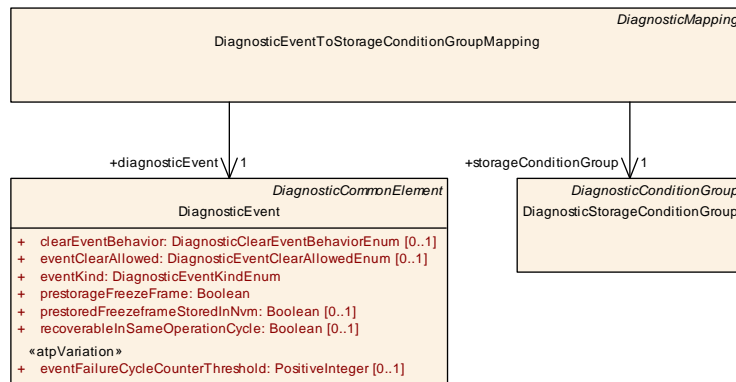


Figure 6.17: DiagnosticEventToStorageConditionGroupMapping

[TPS\_DEXT\_03006] **Values of the individual DiagnosticStorageConditions** [ The values of the individual DiagnosticStorageConditions 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]. ](RS\_DEXT\_00027)

<b>Class</b>	<b>DiagnosticEventToStorageConditionGroupMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
<b>Note</b>	Defines which StorageConditionGroup is applicable for a DiagnosticEvent. <b>Tags:</b> atp.recommendedPackage=DiagnosticMappings			
<b>Base</b>	ARElement, ARObjct, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
diagnosticEvent	DiagnosticEvent	1	ref	Reference to a DiagnosticEvent to which a Storage ConditionGroup is assigned.





Class		DiagnosticEventToStorageConditionGroupMapping		
storageConditionGroup	<a href="#">DiagnosticStorageConditionGroup</a>	1	ref	Reference to a StorageConditionGroup assigned to a DiagnosticEvent.

**Table 6.46: DiagnosticEventToStorageConditionGroupMapping**

### 6.8.6 DiagnosticEvent to Port Mapping

**[TPS\_DEXT\_03007] Semantics of [DiagnosticEventPortMapping](#)** [ A [DiagnosticEventPortMapping](#) defines which [SwcServiceDependencies](#) 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](#)). ]([RS\\_DEXT\\_00023](#), [RS\\_DEXT\\_00052](#))

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](#)** [ If a [DiagnosticEventPortMapping](#) exists and the enclosed [DiagnosticEventPortMapping.diagnosticEvent](#) is also referenced by a [DiagnosticEventToDebounceAlgorithmMapping](#) then the concrete subclass of the respective [DiagnosticEventToDebounceAlgorithmMapping.debounceAlgorithm](#) shall be identical to the [DiagnosticEventPortMapping.swcServiceDependencyInSystem/swcFlatServiceDependency.serviceNeeds.diagEventDebounceAlgorithm](#).

It is assumed that the [DiagnosticEventPortMapping.swcServiceDependencyInSystem/swcFlatServiceDependency.serviceNeeds](#) is a [DiagnosticEventNeeds](#). ]()

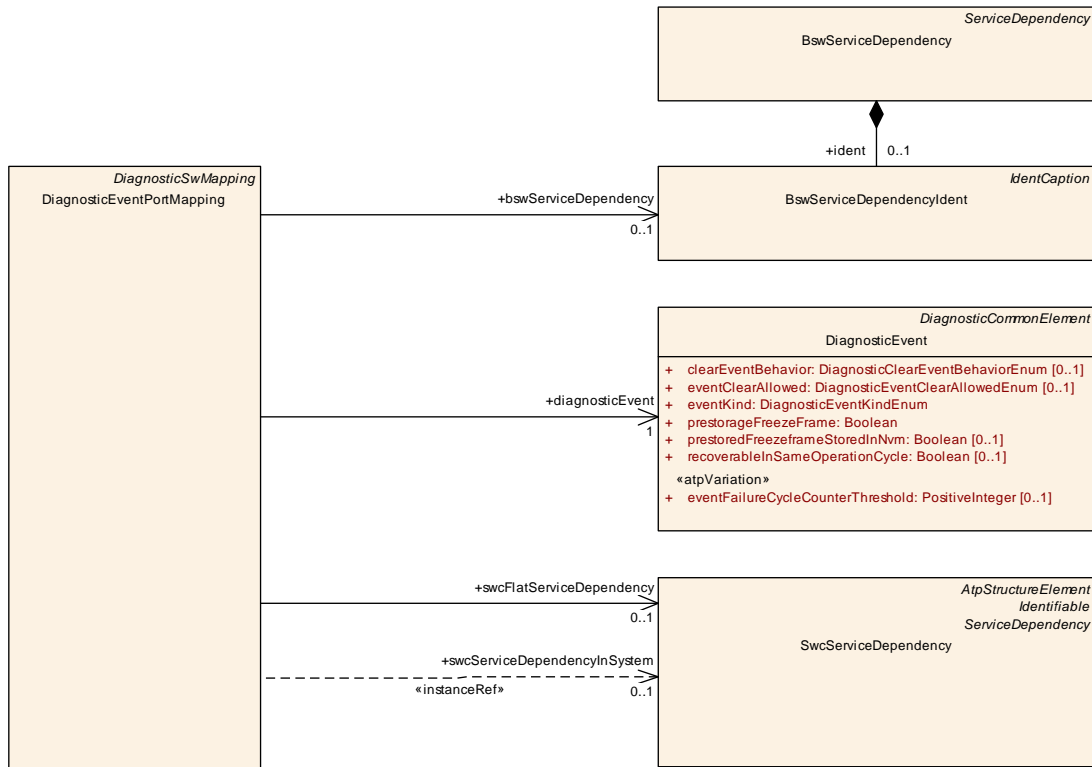


Figure 6.18: DiagnosticEventPortMapping

<b>Class</b>	<b>DiagnosticEventPortMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
<b>Note</b>	Defines to which SWC service ports with DiagnosticEventNeeds the DiagnosticEvent is mapped. <b>Tags:</b> atp.recommendedPackage=DiagnosticMappings			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMapping</a> , <a href="#">DiagnosticSwMapping</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
bswServiceDependency	<a href="#">BswServiceDependencyIdent</a>	0..1	ref	Reference to a BswServiceDependency that links ServiceNeeds to BswModuleEntries.
diagnosticEvent	<a href="#">DiagnosticEvent</a>	1	ref	Reference to the DiagnosticEvent that is assigned to SWC service ports with DiagnosticEventNeeds.
swcFlatServiceDependency	<a href="#">SwcServiceDependency</a>	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports.
swcServiceDependencyInSystem	<a href="#">SwcServiceDependency</a>	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports.

Table 6.47: DiagnosticEventPortMapping

### 6.8.7 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](#).  
|(RS\_DEXT\_00052, RS\_DEXT\_00053)

If such an instance is not yet available, an ordinary reference to [SwcServiceDependency](#) can be given alternatively (i.e. without specifying a certain instance).

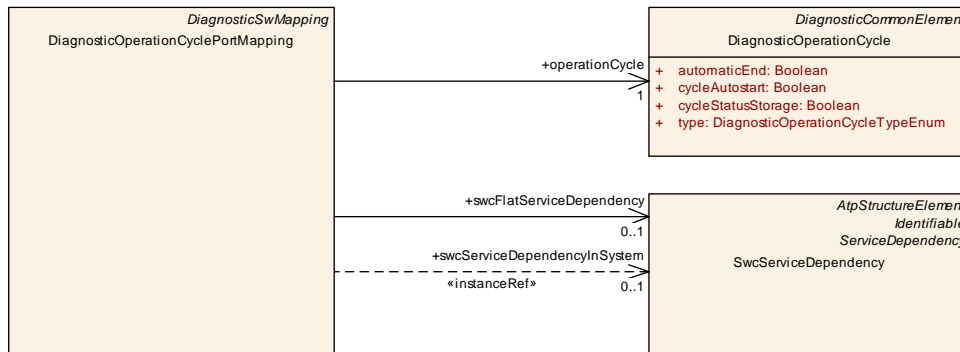


Figure 6.19: DiagnosticOperationCyclePortMapping

<b>Class</b>	<b>DiagnosticOperationCyclePortMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
<b>Note</b>	Defines to which SWC service ports with DiagnosticOperationCycleNeeds the DiagnosticOperationCycle is mapped. <b>Tags:</b> atp.recommendedPackage=DiagnosticMappings			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMapping</a> , <a href="#">DiagnosticSwMapping</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
operationCycle	<a href="#">DiagnosticOperation Cycle</a>	1	ref	Reference to the DiagnosticOperationCycle that is assigned to SWC service ports with DiagnosticOperation CycleNeeds.
swcFlatService Dependency	<a href="#">SwcService Dependency</a>	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports.
swcService DependencyIn System	<a href="#">SwcService Dependency</a>	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports.

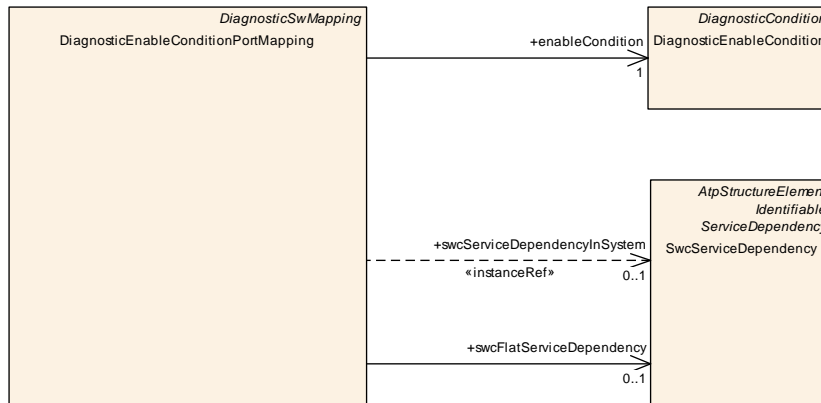
Table 6.48: DiagnosticOperationCyclePortMapping

### 6.8.8 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](#). ]  
(RS\_DEXT\_00026, RS\_DEXT\_00052)

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 6.20: DiagnosticEnableConditionPortMapping**

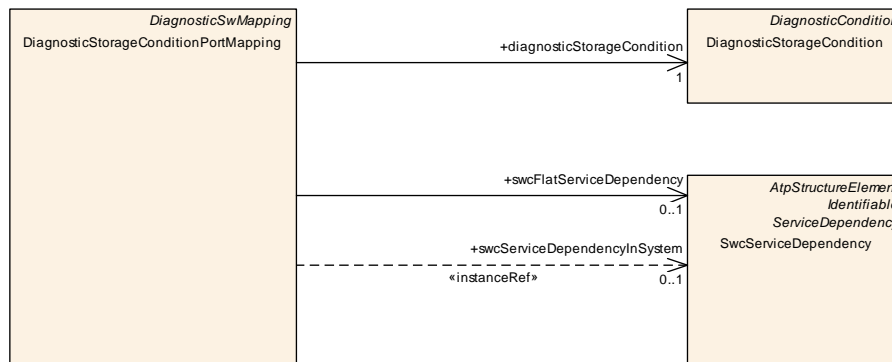
<b>Class</b>	<b>DiagnosticEnableConditionPortMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
<b>Note</b>	Defines to which SWC service ports with DiagnosticEnableConditionNeeds the DiagnosticEnable Condition is mapped. <b>Tags:</b> atp.recommendedPackage=DiagnosticMappings			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMapping</a> , <a href="#">DiagnosticSwMapping</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
enableCondition	<a href="#">DiagnosticEnableCondition</a>	1	ref	Reference to the EnableCondition which is mapped to a SWC service port with DiagnosticEnableConditionNeeds.
swcFlatServiceDependency	<a href="#">SwcServiceDependency</a>	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.
swcServiceDependencyInSystem	<a href="#">SwcServiceDependency</a>	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports.

**Table 6.49: DiagnosticEnableConditionPortMapping**

### 6.8.9 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](#). ] ([RS\\_DEXT\\_00027](#), [RS\\_DEXT\\_00052](#))

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 6.21: DiagnosticStorageConditionPortMapping**

<b>Class</b>	<b>DiagnosticStorageConditionPortMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
<b>Note</b>	Defines to which SWC service ports with DiagnosticStorageConditionNeeds the DiagnosticStorage Condition is mapped. <b>Tags:</b> atp.recommendedPackage=DiagnosticMappings			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMapping</a> , <a href="#">DiagnosticSwMapping</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
diagnostic Storage Condition	<a href="#">DiagnosticStorage Condition</a>	1	ref	Reference to the StorageCondition which is mapped to a SWC service port with DiagnosticStorageCondition Needs.
swcFlatService Dependency	<a href="#">SwcService Dependency</a>	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports.
swcService DependencyIn System	<a href="#">SwcService Dependency</a>	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports.

**Table 6.50: DiagnosticStorageConditionPortMapping**

### 6.8.10 Provided Data Mapping

[TPS\_DEXT\_03020] Semantics of [DiagnosticDemProvidedDataMapping](#) [ The meta-class [DiagnosticDemProvidedDataMapping](#) does not seem to fulfill the condition for representing a mapping class because it only has one reference to a [DiagnosticDataElement](#) in the role `dataElement`.

However, the specific nature of this mapping is that the second element (the [DiagnosticDemProvidedDataMapping.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 [DiagnosticDemProvidedDataMapping.dataProvider](#) by a `NameToken`.

The semantics of this mapping is to further qualify the access to the [DiagnosticDataElement](#) referenced in the role `dataElement` from within the Dem. ] ([RS\\_DEXT\\_00043](#), [RS\\_DEXT\\_00052](#))

[TPS\_DEXT\_01142] **Standardized values for [DiagnosticDemProvidedDataMapping.dataProvider](#)** [ For attribute [DiagnosticDemProvidedDataMapping.dataProvider](#), the following values are reserved by the AUTOSAR standard:

- **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\_SINCE\_FIRST\_FAILED**: map Dem-internal Operation Cycle Counter - cycles since first failed.
- **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.

]([RS\\_DEXT\\_00043](#), [RS\\_DEXT\\_00052](#))

Please note that, where applicable, AUTOSAR allows for the usage of values of the attribute [DiagnosticDemProvidedDataMapping.dataProvider](#) other than the values standardized by [TPS\_DEXT\_01142].

In this case, however, proprietary values of the attribute `DiagnosticDemProvided-DataMapping.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.

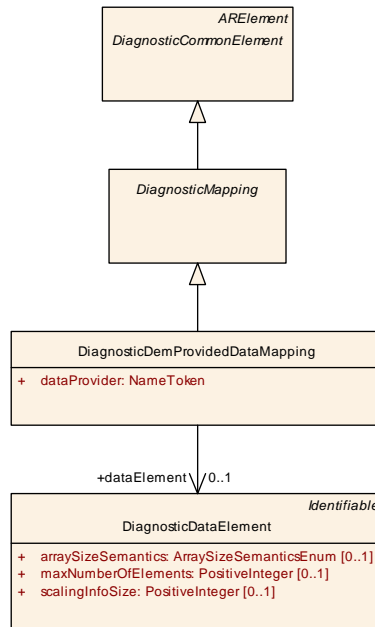


Figure 6.22: Modeling of the `DiagnosticDemProvidedDataMapping`

<b>Class</b>	<b>DiagnosticDemProvidedDataMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::ServiceMapping			
<b>Note</b>	This represents the ability to define the nature of a data access for a <code>DiagnosticDataElement</code> in the Dem. <b>Tags:</b> <code>atp.recommendedPackage=DiagnosticServiceMappings</code>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMapping</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataElement	<a href="#">DiagnosticDataElement</a>	0..1	ref	This represents the <code>DiagnosticDataElement</code> for which the access is further qualified by the <code>DiagnosticDemProvidedDataMapping</code> .
dataProvider	<a href="#">NameToken</a>	1	attr	This represents the ability to further specify the access within the Dem.

Table 6.51: `DiagnosticDemProvidedDataMapping`

### 6.8.11 Master to Slave Event Mapping

AUTOSAR provides the ability to associate multiple diagnostic events in the way 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.

Please note that 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`** [ Any given `DiagnosticEvent` shall at most once be referenced 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** [ A `DiagnosticEvent` referenced in the role `DiagnosticMasterToSlaveEventMapping.slaveEvent` shall not be referenced in the role `DiagnosticEventPortMapping.diagnosticEvent` and vice versa. ]()

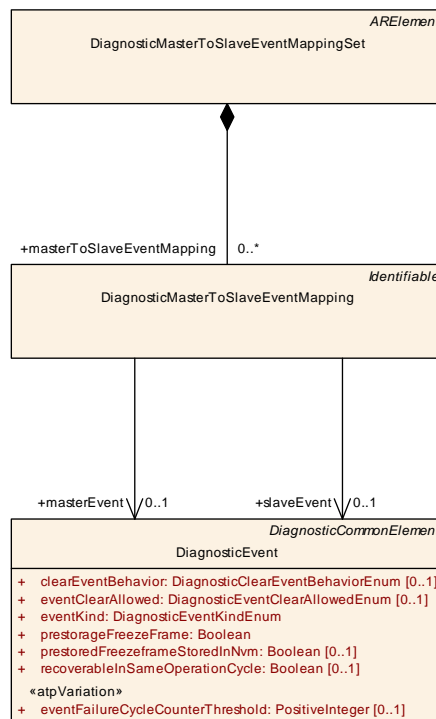


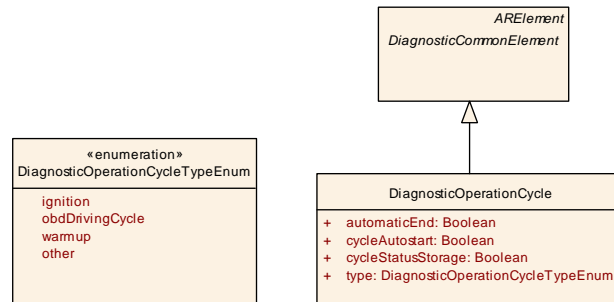
Figure 6.23: Modeling of the `DiagnosticMasterToSlaveEventMapping`

<b>Class</b>	<b>DiagnosticMasterToSlaveEventMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
<b>Note</b>	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			
<b>Base</b>	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
masterEvent	<a href="#">DiagnosticEvent</a>	0..1	ref	This represents the master diagnostic event.
slaveEvent	<a href="#">DiagnosticEvent</a>	0..1	ref	This represents the slave diagnostic event.

**Table 6.52: DiagnosticMasterToSlaveEventMapping**

## 6.9 DiagnosticOperationCycle

[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. ] ([RS\\_DEXT\\_00054](#))



**Figure 6.24: Modeling of DiagnosticOperationCycle**

<b>Class</b>	<b>DiagnosticOperationCycle</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticOperationCycle			
<b>Note</b>	Definition of an operation cycle that is the base of the event qualifying and for Dem scheduling. <b>Tags:</b> atp.recommendedPackage=DiagnosticOperationCycles			
<b>Base</b>	<i>ARElement</i> , <i>ARObject</i> , <i>CollectableElement</i> , <i>DiagnosticCommonElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
automaticEnd	Boolean	1	attr	If set to true the driving cycle shall automatically end at either Dem_Shutdown() or Dem_Init().
cycleAutostart	Boolean	1	attr	This attribute defines if the operation cycles is automatically re-started during Dem_PreInit.
cycleStatus Storage	Boolean	1	attr	Defines if the operation cycle state is available over the power cycle (stored non-volatile) or not. <ul style="list-style-type: none"> <li>• true: the operation cycle state is stored non-volatile</li> <li>• false: the operation cycle state is only stored volatile</li> </ul>





Class	DiagnosticOperationCycle			
type	<a href="#">DiagnosticOperationCycleTypeEnum</a>	1	attr	Operation cycles types for the Dem.

**Table 6.53: DiagnosticOperationCycle**

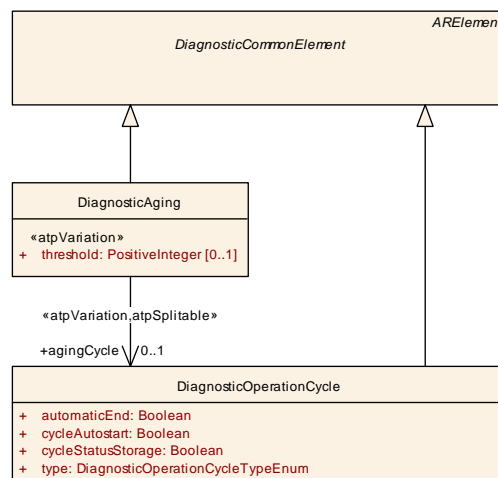
Enumeration	DiagnosticOperationCycleTypeEnum
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticOperationCycle
Note	Operation cycles types used to identify certain Operation cycles with a certain semantics.
Literal	Description
ignition	Ignition ON / OFF cycle <b>Tags:</b> atp.EnumerationValue=0
obdDrivingCycle	OBD Driving cycle <b>Tags:</b> atp.EnumerationValue=1
other	further operation cycle <b>Tags:</b> atp.EnumerationValue=2
warmup	OBD Warm up cycle <b>Tags:</b> atp.EnumerationValue=5

**Table 6.54: DiagnosticOperationCycleTypeEnum**

## 6.10 DiagnosticAging

[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](#). ]  
(RS\_DEXT\_00055)



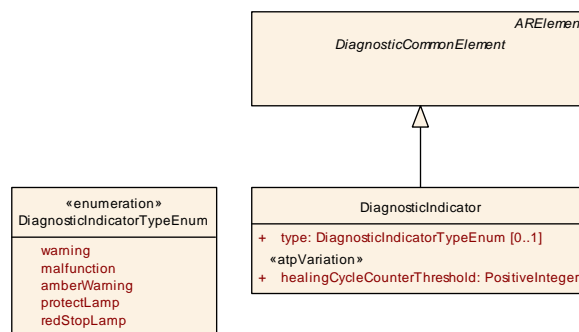
**Figure 6.25: Modeling of DiagnosticAging**

<b>Class</b>	<b>DiagnosticAging</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticAging			
<b>Note</b>	Defines the aging algorithm. <b>Tags:</b> atp.recommendedPackage=DiagnosticAgings			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">Multilanguage</a> , <a href="#">Referrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
agingCycle	<a href="#">DiagnosticOperation Cycle</a>	0..1	ref	This represents the applicable aging cycle. <b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=agingCycle, variationPoint.ShortLabel vh.latestBindingTime=preCompileTime
threshold	PositiveInteger	0..1	attr	Number of aging cycles needed to unlearn/delete the event. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime

**Table 6.55: DiagnosticAging**

## 6.11 DiagnosticIndicator

[TPS\_DEXT\_03022] Different kinds of **DiagnosticIndicators** [ Different *types* of Indicators can be defined with the [DiagnosticIndicator](#) element. For this, the attribute [DiagnosticIndicator.type](#) shall be used. ] ([RS\\_DEXT\\_00056](#))



**Figure 6.26: Modeling of DiagnosticIndicator**

<b>Class</b>	<b>DiagnosticIndicator</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticIndicator			
<b>Note</b>	Definition of an indicator. <b>Tags:</b> atp.recommendedPackage=DiagnosticIndicators			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">Multilanguage</a> , <a href="#">Referrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
healingCycle Counter Threshold	PositiveInteger	1	attr	This attribute defines the number of healing cycles for the WarningIndicatorOffCriteria <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime







<b>Class</b>	<b>DiagnosticIndicator</b>			
type	<a href="#">DiagnosticIndicatorTypeEnum</a>	0..1	attr	Defines the type of the indicator.

**Table 6.56: DiagnosticIndicator**

<b>Enumeration</b>	<b>DiagnosticIndicatorTypeEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticIndicator
<b>Note</b>	Type of an indicator.
<b>Literal</b>	<b>Description</b>
amberWarning	Amber Warning Lamp <b>Tags:</b> atp.EnumerationValue=0
malfunction	Malfunction Indicator Lamp <b>Tags:</b> atp.EnumerationValue=1
protectLamp	Protect Lamp <b>Tags:</b> atp.EnumerationValue=2
redStopLamp	Red Stop Lamp <b>Tags:</b> atp.EnumerationValue=3
warning	Warning <b>Tags:</b> atp.EnumerationValue=4

**Table 6.57: DiagnosticIndicatorTypeEnum**

## 6.12 DiagnosticTestResult

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 latest test results back to a client. This is of special importance for the OBD service mode 0x06, see chapter [5.6.5](#).

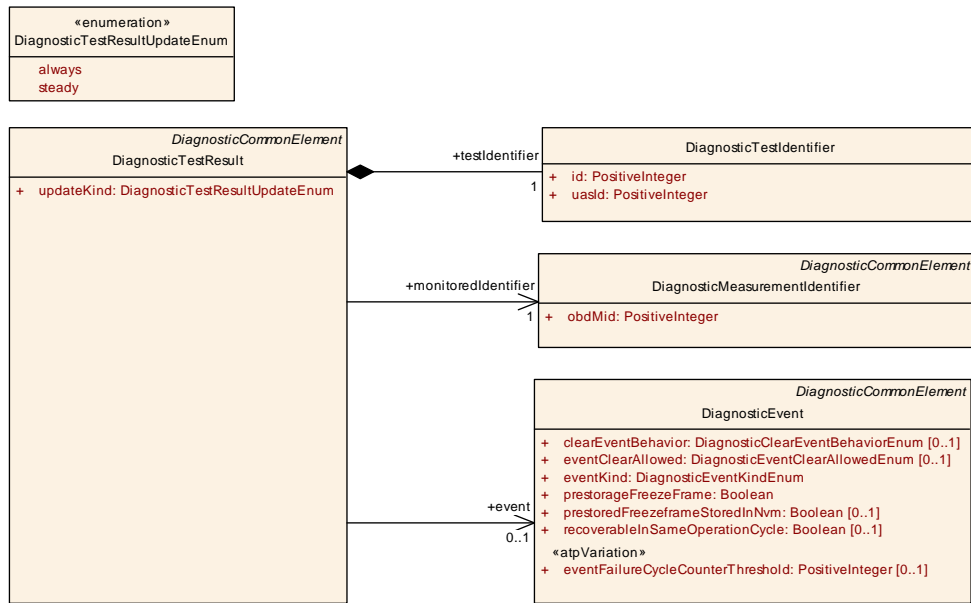


Figure 6.27: Modeling of DiagnosticTestResult

<b>Class</b>	<b>DiagnosticTestResult</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTestResult			
<b>Note</b>	This meta-class represents the ability to define diagnostic test results. <b>Tags:</b> atp.recommendedPackage=DiagnosticTestResults			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
event	<a href="#">DiagnosticEvent</a>	0..1	ref	This attribute represents the diagnostic event that is related to the diagnostic test result.
monitored Identifier	<a href="#">DiagnosticMeasurementIdentifier</a>	1	ref	This attribute represents the related diagnostic monitored identifier.
testIdentifier	<a href="#">DiagnosticTestIdentifier</a>	1	aggr	This attribute represents the applicable test identifier.
updateKind	<a href="#">DiagnosticTestResultUpdateEnum</a>	1	attr	This attribute controls the update behavior of the enclosing DiagnosticTestResult.

Table 6.58: DiagnosticTestResult

<b>Enumeration</b>	<b>DiagnosticTestResultUpdateEnum</b>	
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTestResult	
<b>Note</b>	This meta-class represents the ability to define the update behavior of a DiagnosticTestResult.	
<b>Literal</b>	<b>Description</b>	
always	Any DTR result reported by the monitor is used by the Dem. <b>Tags:</b> atp.EnumerationValue=0	
steady	The Dem accepts reported DTRs only when the configured debouncing mechanism is stable at the FAIL or PASS limit. <b>Tags:</b> atp.EnumerationValue=1	

Table 6.59: DiagnosticTestResultUpdateEnum

<b>Class</b>	<b>DiagnosticTestIdentifier</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTestResult			
<b>Note</b>	This meta-class represents the ability to create a diagnostic test identifier.			
<b>Base</b>	<i>ARObject</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
id	PositiveInteger	1	attr	This represents the numerical id associated with the diagnostic test identifier.
uasId	PositiveInteger	1	attr	This represents the unit and scaling Id of the diagnostic test result.

**Table 6.60: DiagnosticTestIdentifier**

<b>Class</b>	<b>DiagnosticMeasurementIdentifier</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTestResult			
<b>Note</b>	This meta-class represents the ability to describe a measurement identifier. <b>Tags:</b> atp.recommendedPackage=DiagnosticMeasurementIdentifiers			
<b>Base</b>	<i>ARElement</i> , <i>ARObject</i> , <i>CollectableElement</i> , <i>DiagnosticCommonElement</i> , <i>Identifiable</i> , <i>Multilanguage</i> , <i>Referrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
obdMid	PositiveInteger	1	attr	This represents the numerical measurement Id

**Table 6.61: DiagnosticMeasurementIdentifier**

## 6.13 OBD-related aspects of Dem Configuration

The support for OBD-related [18] modeling requires the addition of some pretty global (i.e. on the level of an entire ECU) attribute 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 where an actual [EcuInstance](#) is not yet available.

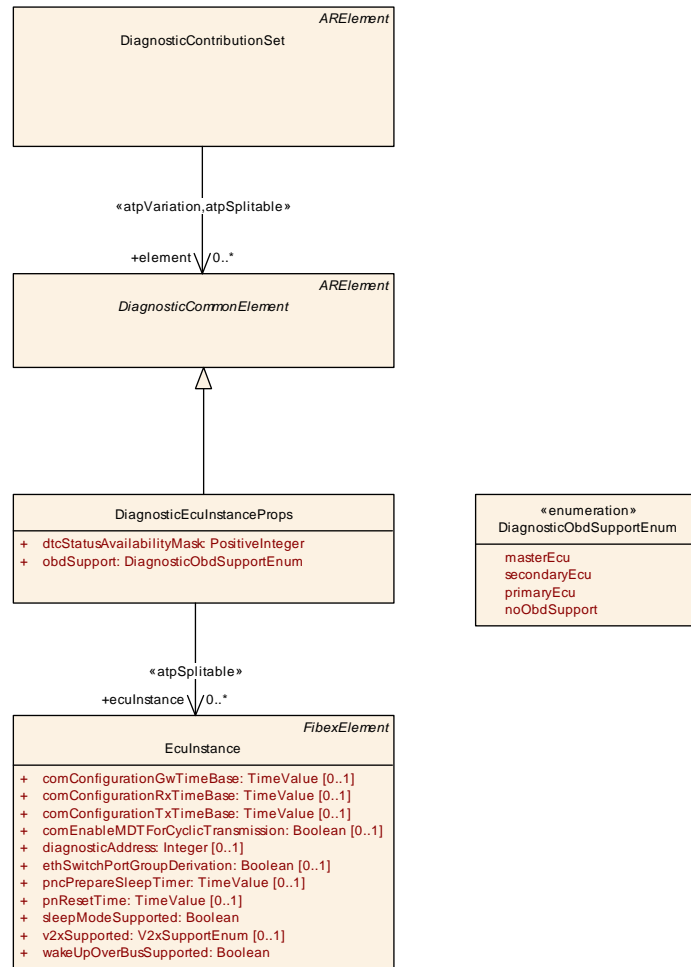


Figure 6.28: Modeling of **DiagnosticEcuInstanceProps**

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 or not a given **EcuInstance** shall support OBD and in which way OBD shall be supported on this Ecu. ] (**RS\_DEXT\_00058**)

<b>Class</b>	<b>DiagnosticEcuInstanceProps</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticContribution
<b>Note</b>	<p>This meta-class represents the ability to model properties that are specific for a given <b>EcuInstance</b> but on the other hand represent purely diagnostic-related information.</p> <p>In the spirit of decentralized configuration it is therefore possible to specify the diagnostic-related information related to a given <b>EcuInstance</b> even if the <b>EcuInstance</b> does not yet exist.</p> <p><b>Tags:</b> atp.recommendedPackage=DiagnosticEcuInstancePropss</p>
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">AObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">Multilanguage</a> , <a href="#">Referrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>





Class		DiagnosticEcuInstanceProps		
Attribute	Type	Mul.	Kind	Note
dtcStatus AvailabilityMask	PositiveInteger	1	attr	This attribute contains the value of the DTC status availability mask.
ecuInstance	<a href="#">EcuInstance</a>	*	ref	This represents the actual EcuInstance to which the information contained in the DiagnosticEcuInstance contribute.  <b>Stereotypes:</b> atpSplittable <b>Tags:</b> atp.Splitkey=ecuInstance
obdSupport	<a href="#">DiagnosticObdSupportEnum</a>	1	attr	This attribute is used to specify the role (if applicable) in which the DiagnosticEcuInstance supports OBD.

**Table 6.62: DiagnosticEcuInstanceProps**

Enumeration	DiagnosticObdSupportEnum
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticContribution
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.
Literal	Description
masterEcu	This represent the role "master ECU". <b>Tags:</b> atp.EnumerationValue=0
noObdSupport	This represents the ability to explicitly specify that no participation in OBD is foreseen. <b>Tags:</b> atp.EnumerationValue=1
primaryEcu	This represents the role "primary ECU". <b>Tags:</b> atp.EnumerationValue=2
secondaryEcu	This represents the role "secondary ECU". <b>Tags:</b> atp.EnumerationValue=3

**Table 6.63: 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 on 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\_01110] Standardized values of **DiagnosticIumprGroup.category**

[ The semantics of the **DiagnosticIumprGroup** are identified by means of the attribute **DiagnosticIumprGroup.category**. Standardized values of **DiagnosticIumprGroup.category** are:

- IUMPR\_DENOMINATOR\_NONE
- IUMPR\_DENOMINATOR\_PHYSICAL\_API
- IUMPR\_DENOMINATOR\_COLDSTART
- IUMPR\_DENOMINATOR\_EVAP
- IUMPR\_DENOMINATOR\_500\_MILES

](RS\_DEXT\_00078)

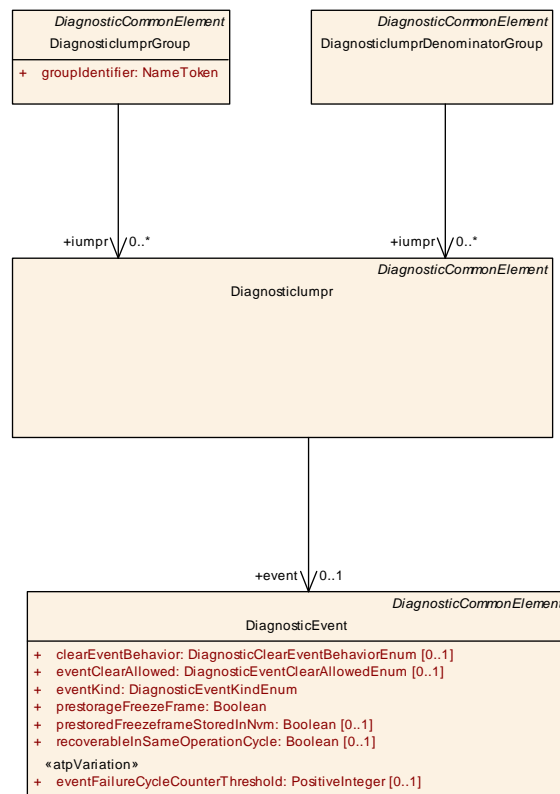


Figure 6.29: Modeling of IUMPR handling for diagnostic events

<b>Class</b>	<b>DiagnosticIumpr</b>
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent
<b>Note</b>	This meta-class represents the ability to model the in-us 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. <b>Tags:</b> atp.recommendedPackage=DiagnosticIumpr





<b>Class</b>	<b>Diagnosticlumpr</b>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
event	<a href="#">DiagnosticEvent</a>	0..1	ref	This reference represents the DiagnosticEvent that corresponds to the IUMPR computation.

**Table 6.64: Diagnosticlumpr**

<b>Class</b>	<b>DiagnosticlumprGroup</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
<b>Note</b>	This meta-class represents the ability to model a IUMPR groups. <b>Tags:</b> atp.recommendedPackage=DiagnosticlumprGroups			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
groupIdentifier	<a href="#">NameToken</a>	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.
iumpr	<a href="#">Diagnosticlumpr</a>	*	ref	This reference collects Diagnosticlumpr to a Diagnostic lumprGroup.

**Table 6.65: DiagnosticlumprGroup**

<b>Class</b>	<b>DiagnosticlumprDenominatorGroup</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
<b>Note</b>	This meta-class represents the ability to model a IUMPR denominator groups. <b>Tags:</b> atp.recommendedPackage=DiagnosticlumprDenominatorGroup			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
iumpr	<a href="#">Diagnosticlumpr</a>	*	ref	This reference collects Diagnosticlumpr to a Diagnostic lumprDenominatorGroup.

**Table 6.66: DiagnosticlumprDenominatorGroup**

### 6.13.1 Dem Configuration for OBD-II

The modeling of DTCs for the OBD-II use case is pretty similar to the modeling of DTCs for the UDS use case. In other words, [DiagnosticTroubleCodeObd](#) references the [DiagnosticTroubleCodeProps](#) in the same way that this meta-class is referenced from [DiagnosticTroubleCodeUds](#).

Please note that the meta-class [DiagnosticTroubleCodeObd](#) is only applicable for the implementation of 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`. ]()

**[constr\_1459] Existence of attributes of `DiagnosticTroubleCodeProps`** [ The following list of attributes of meta-class `DiagnosticTroubleCodeProps` are not required and therefore shall be ignored if the `DiagnosticTroubleCodeProps` is referenced in the role `dtcProps` from a `DiagnosticTroubleCodeObd`:

- `freezeFrame`
- `freezeFrameContent`
- `memoryDestination`
- `extendedDataRecord`
- `aging`

]()



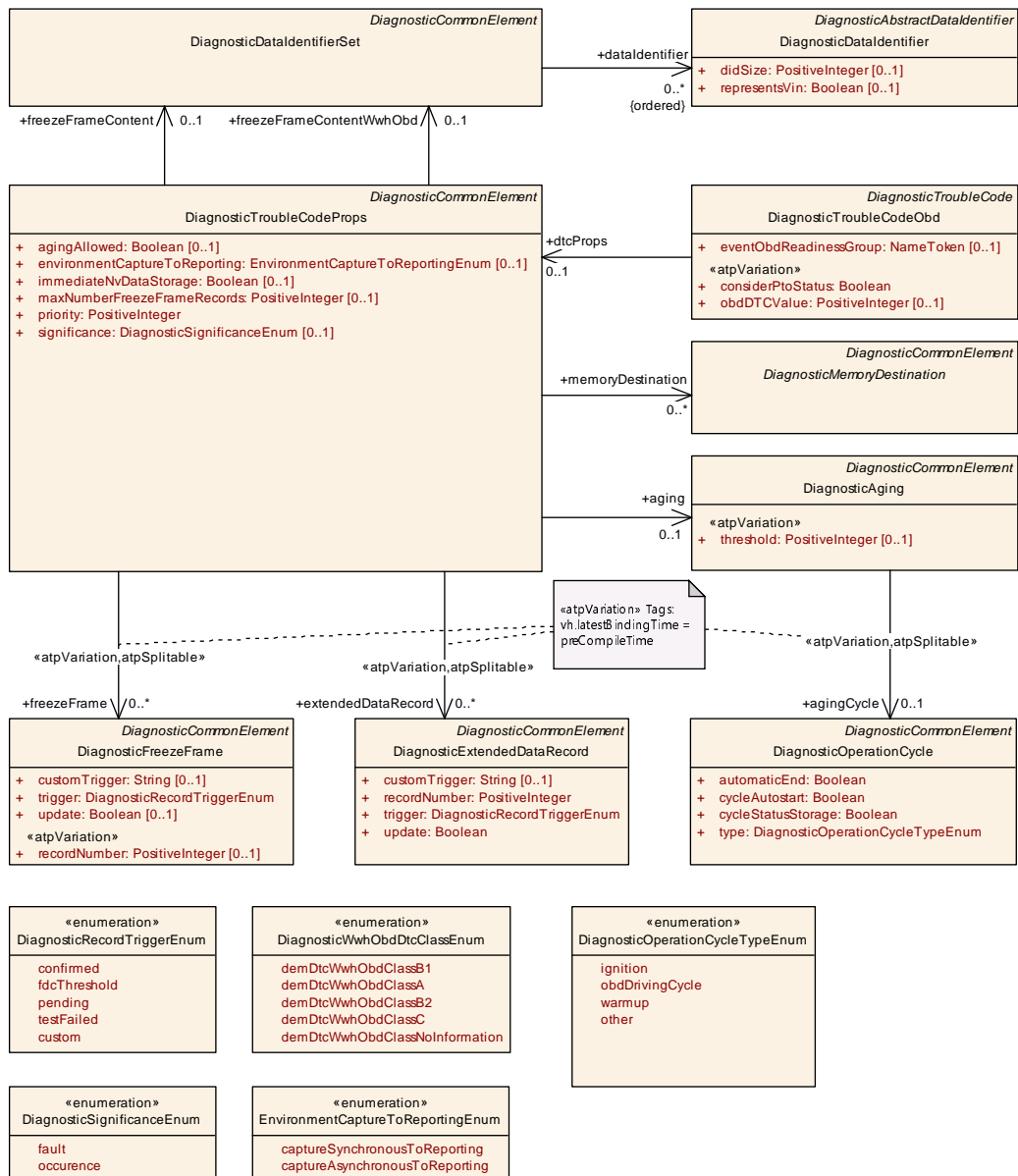


Figure 6.30: Modeling of DTCs for OBD-II

### 6.13.2 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 `whObdDtcClass` 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 `DiagnosticTroubleCode-Props.extendedDataRecord` with attribute `recordNumber` set to the value 90 is supported. `]()`

## 7 Functional Inhibition

### 7.1 Introduction

Conceptually, the Fim [14] 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`.

### 7.2 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.

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`.

<b>Class</b>	<b>DiagnosticFimAliasEvent</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
<b>Note</b>	<p>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 <code>DiagnosticFimAliasEvent</code> has a reference to the actual <code>DiagnosticEvent</code> and by this the final connection is created.</p> <p><b>Tags:</b> atp.recommendedPackage=DiagnosticFimAliasEvents</p>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticAbstractAliasEvent</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 7.1: DiagnosticFimAliasEvent**

<b>Class</b>	<i>DiagnosticAbstractAliasEvent</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
<b>Note</b>	This meta-class represents an abstract base class for all diagnostic alias events.			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">DiagnosticFimAliasEvent</a> , <a href="#">DiagnosticFimAliasEventGroup</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 7.2: DiagnosticAbstractAliasEvent**

### 7.3 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.

] ([RS\\_DEXT\\_00060](#))

Please note that [DiagnosticEvent](#) is discussed in chapter 6.2.

<b>Class</b>	<b>DiagnosticFunctionIdentifier</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
<b>Note</b>	This meta-class represents a diagnostic function identifier (a.k.a. FID). <b>Tags:</b> atp.recommendedPackage=DiagnosticFunctionIdentifiers			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			





Class	DiagnosticFunctionIdentifier			
Attribute	Type	Mul.	Kind	Note
-	-	-	-	-

**Table 7.3: DiagnosticFunctionIdentifier**

Finally all sources will refer to [DiagnosticEvents](#).

## 7.4 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 many diagnostic events (modeled as [DiagnosticEvent](#)).

The easiest way to provide this information is the usage of the [DiagnosticInhibitSourceEventMapping](#) that is able to create an association between a [DiagnosticInhibitSourceEventMapping](#) 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 may or may not apply. For more details, please refer to [Figure 7.1](#).

Class	DiagnosticInhibitSourceEventMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
Note	<p>This meta-class represents the ability to map a <a href="#">DiagnosticFunctionInhibitSource</a> directly to alternatively one <a href="#">DiagnosticEvent</a> or one <a href="#">DiagnosticFimSummaryEvent</a>. 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.</p> <p><b>Tags:</b> atp.recommendedPackage=DiagnosticInhibitSourceEventMappings</p>			
Base	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMapping</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
Attribute	Type	Mul.	Kind	Note
diagnosticEvent	<a href="#">DiagnosticEvent</a>	0..1	ref	This represents the reference to the diagnostic event.
eventGroup	<a href="#">DiagnosticFimEventGroup</a>	0..1	ref	This represents the reference to the event group
inhibitionSource	<a href="#">DiagnosticFunctionInhibitSource</a>	0..1	ref	This represents the reference to the inhibition source.

**Table 7.4: DiagnosticInhibitSourceEventMapping**

## 7.5 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](#)

|(RS\_DEXT\_00061, RS\_DEXT\_00062)

<b>Class</b>	<b>DiagnosticFimAliasEventMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
<b>Note</b>	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. <b>Tags:</b> atp.recommendedPackage=DiagnosticFimEventMappings			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMapping</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
actualEvent	<a href="#">DiagnosticEvent</a>	0..1	ref	This represents the reference to the actual diagnostic event.
aliasEvent	<a href="#">DiagnosticFimAliasEvent</a>	0..1	ref	This represents the reference to the alias event.

**Table 7.5: DiagnosticFimAliasEventMapping**

<b>Class</b>	<b>DiagnosticFimAliasEventGroup</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
<b>Note</b>	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. <b>Tags:</b> atp.recommendedPackage=DiagnosticFimAliasEventGroups			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticAbstractAliasEvent</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
groupedAliasEvent	<a href="#">DiagnosticFimAliasEvent</a>	*	ref	By means of this reference the grouping of Diagnostic AliasEvents within the DiagnosticFimSummaryEvent can be specified.

**Table 7.6: DiagnosticFimAliasEventGroup**

<b>Class</b>	<b>DiagnosticFimAliasEventGroupMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
<b>Note</b>	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. <b>Tags:</b> atp.recommendedPackage=DiagnosticFimAliasEventGroupMappings			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMapping</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
actualEvent	<a href="#">DiagnosticFimEventGroup</a>	0..1	ref	This represents the reference to the actual summary event.





<i>Class</i>	<b>DiagnosticFimAliasEventGroupMapping</b>			
aliasEvent	<a href="#">DiagnosticFimAliasEventGroup</a>	0..1	ref	This represents the reference to the alias summary event.

**Table 7.7: DiagnosticFimAliasEventGroupMapping**

Ultimately, the modeling approach for the Fim starts at the definition of the concept of a function itself by means of the meta-class [DiagnosticFunctionIdentifier](#).

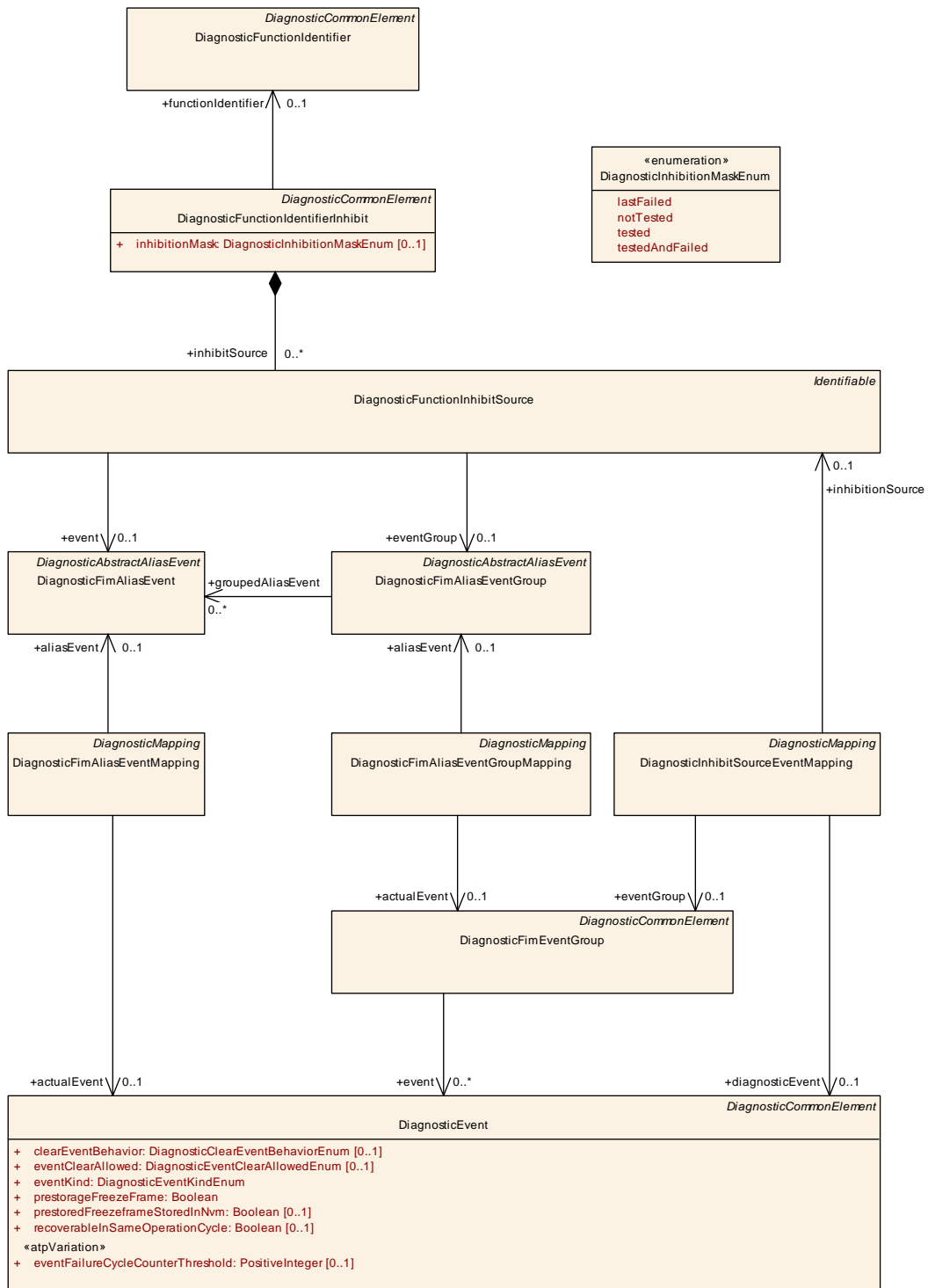


Figure 7.1: Big Picture of Fim diagnostics configuration

<b>Class</b>	<b>DiagnosticFunctionIdentifierInhibit</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
<b>Note</b>	This meta-class represents the ability to define the inhibition of a specific function identifier within the Fim configuration. <b>Tags:</b> atp.recommendedPackage=DiagnosticFunctionIdentifierInhibits			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
functionIdentifier	<a href="#">DiagnosticFunctionIdentifier</a>	0..1	ref	This represents the corresponding function identifier.
inhibitSource	<a href="#">DiagnosticFunctionInhibitSource</a>	*	aggr	This represents a collection of DiagnosticFunctionInhibitSource that contribute to the configuration of the enclosing DiagnosticFunctionIdentifierInhibit.
inhibitionMask	<a href="#">DiagnosticInhibitionMaskEnum</a>	0..1	attr	This represents the value of the inhibition mask behavior.

**Table 7.8: DiagnosticFunctionIdentifierInhibit**

<b>Class</b>	<b>DiagnosticFunctionInhibitSource</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
<b>Note</b>	This meta-class represents the ability to define an inhibition source in the context of the Fim configuration.			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
event	<a href="#">DiagnosticFimAliasEvent</a>	0..1	ref	This represents the alias event applicable for the referencing inhibition source.
eventGroup	<a href="#">DiagnosticFimAliasEventGroup</a>	0..1	ref	This represents the event group applicable for the referencing inhibition source.

**Table 7.9: DiagnosticFunctionInhibitSource**

<b>Enumeration</b>	<b>DiagnosticInhibitionMaskEnum</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
<b>Note</b>	This meta-class represents the ability to define different kinds of inhibition mask behavior.			
<b>Literal</b>	<b>Description</b>			
lastFailed	This represents the inhibition mask behavior "last failed". <b>Tags:</b> atp.EnumerationValue=0			
notTested	This represents the inhibition mask behavior "not tested". <b>Tags:</b> atp.EnumerationValue=1			
tested	This represents the inhibition mask behavior "tested". <b>Tags:</b> atp.EnumerationValue=3			
testedAndFailed	This represents the inhibition mask behavior "tested and failed". <b>Tags:</b> atp.EnumerationValue=2			

**Table 7.10: 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. ]  
([RS\\_DEXT\\_00060](#))

[[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`. ]([RS\\_DEXT\\_00060](#))

[[constr\\_1453](#)] References from `DiagnosticFunctionInhibitSource` [ Each `DiagnosticFunctionInhibitSource` may either reference one of the following meta-classes in their respective roles:

- `DiagnosticFimAliasEventMapping` in the role `event`
- `DiagnosticFimAliasEventGroupMapping` in the role `eventGroup`

]()

<b>Class</b>	<b>DiagnosticFimEventGroup</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
<b>Note</b>	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. <b>Tags:</b> atp.recommendedPackage=DiagnosticFimEventGroups			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
event	<a href="#">DiagnosticEvent</a>	*	ref	This reference represents the way of grouping diagnostic events into a summary event in the context of the Fim.

**Table 7.11: DiagnosticFimEventGroup**

[[TPS\\_DEXT\\_01098](#)] Semantics of attribute `DiagnosticFunctionInhibitSource.event` [ If the reference `DiagnosticFunctionInhibitSource.event` exists this means the inhibition applies for a single `DiagnosticEvent` ]  
([RS\\_DEXT\\_00060](#), [RS\\_DEXT\\_00061](#))

[[TPS\\_DEXT\\_01099](#)] Semantics of attribute `DiagnosticFunctionInhibitSource.eventGroup` [ If the reference `DiagnosticFunctionInhibitSource.eventGroup` exists this means that the inhibition effectively applies for a group of `DiagnosticEvents` where the actual grouping is defined in the scope of the Fim. ]([RS\\_DEXT\\_00060](#), [RS\\_DEXT\\_00061](#))

[[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 directly be taken 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.

|(RS\_DEXT\_00060, RS\_DEXT\_00061)

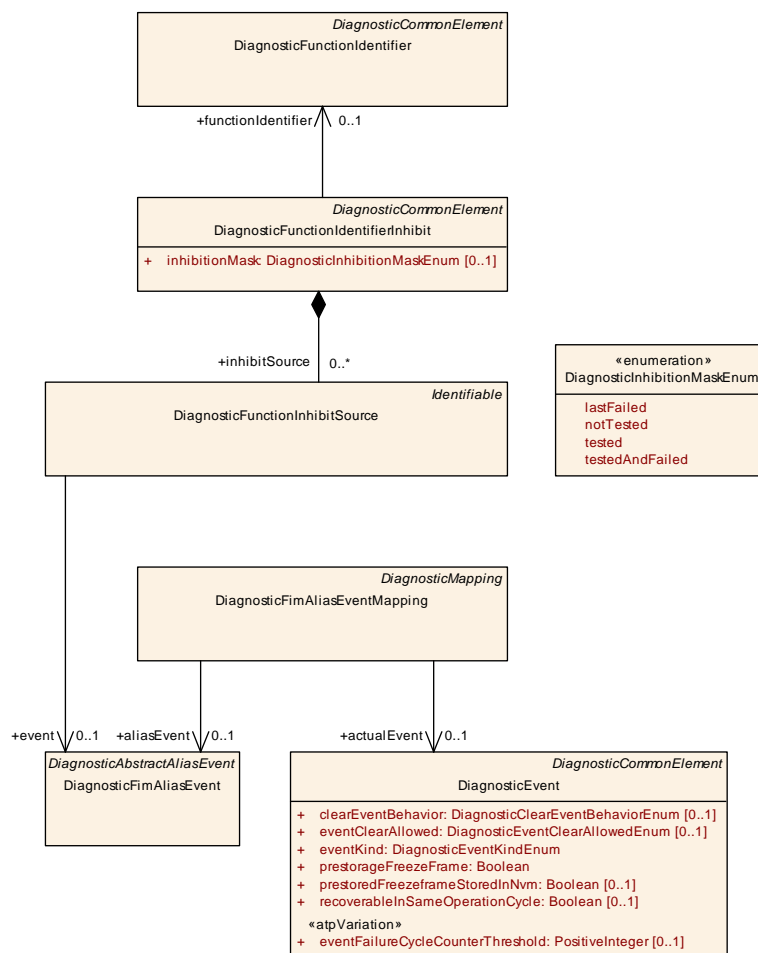


Figure 7.2: Inhibition of events in Fim for diagnostics configuration

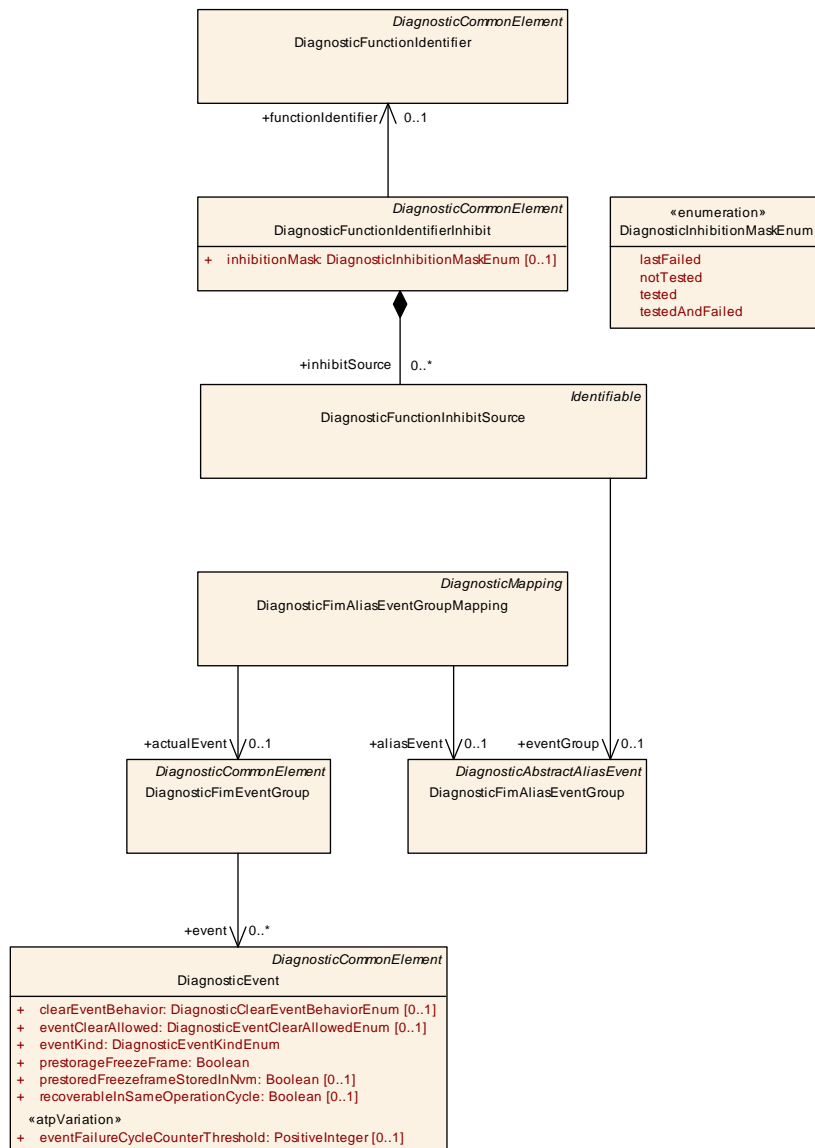
[TPS\_DEXT\_01101] **Consequence of the existence of `DiagnosticFimAliasEventGroupMapping`** | The references from meta-class `DiagnosticFimAliasEventGroupMapping` in the roles `actualEvent` and `aliasEvent` do not necessarily have to exist at the same time. The following rules apply:

**Only actualEvent exists:** In this case there is no DiagnosticFimAliasEventGroup available and the actual DiagnosticFimEventGroup can directly be taken 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.

|(RS\_DEXT\_00060, RS\_DEXT\_00061)



**Figure 7.3: Inhibition of event groups in Fim for diagnostics configuration**

## 7.6 Mapping of Function Identifiers to the corresponding Monitors

Beside the ability to model a function inhibition itself, it is equally important that this inhibition can be connected to the applicable monitors.

This relation can be expressed by means of the `DiagnosticFimFunctionMapping`. The details are sketched in Figure 7.4.

**[TPS\_DEXT\_01102] Semantics of `DiagnosticFimFunctionMapping`** [ The meta-class `DiagnosticFimFunctionMapping` represents the ability to map a `DiagnosticFunctionIdentifier` to a `SwcServiceDependency` that addresses function inhibition. ](*RS\_DEXT\_00063*)

**[constr\_1454] `DiagnosticFimFunctionMapping` shall only reference a `SwcServiceDependency` that aggregates `FunctionInhibitionNeeds`** [ A `DiagnosticFimFunctionMapping` shall only reference a `SwcServiceDependency` that aggregates `FunctionInhibitionNeeds` in the role `serviceNeeds`. ]()

The modeling of this mapping is closely related to the existing modeling of mappings that involve `SwcServiceDependency` in the context of the diagnostic extract.

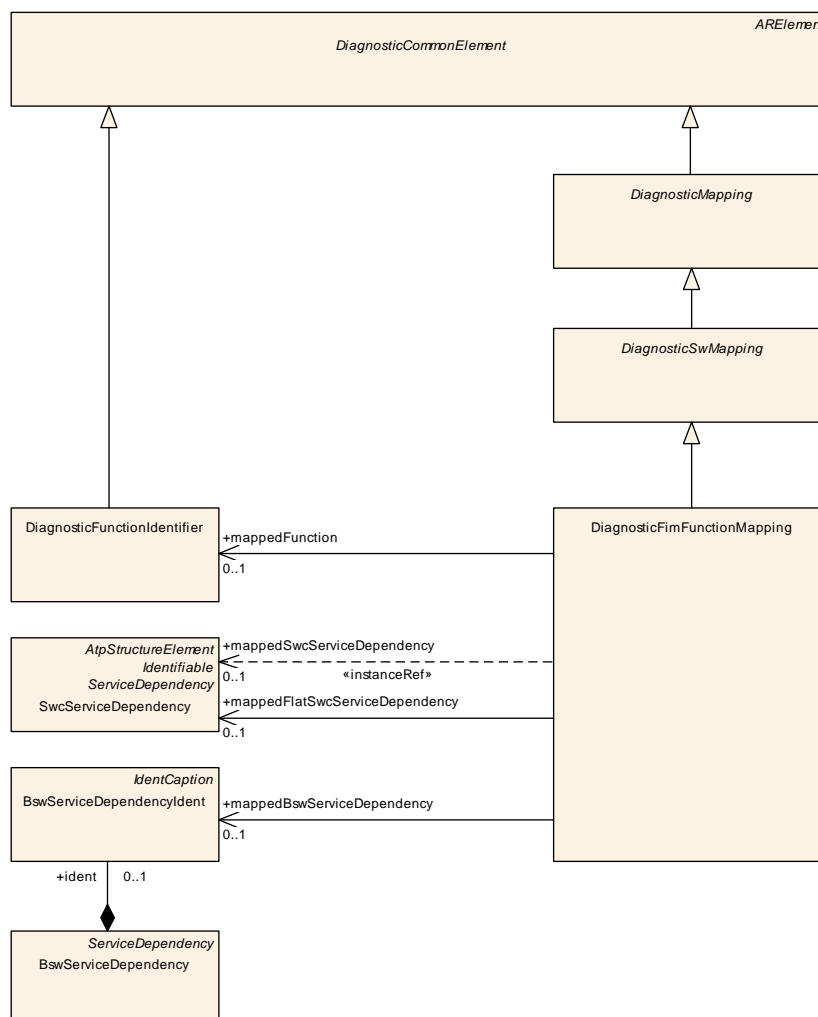


Figure 7.4: Mapping of function identifiers to the corresponding monitors

<b>Class</b>	<b>DiagnosticFimFunctionMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::ServiceMapping			
<b>Note</b>	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. <b>Tags:</b> atp.recommendedPackage=DiagnosticFimFunctionMappings			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMapping</a> , <a href="#">DiagnosticSwMapping</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
mappedBsw Service Dependency	<a href="#">BswServiceDependencyIdent</a>	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.
mappedFlatSwc Service Dependency	<a href="#">SwcServiceDependency</a>	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.
mapped Function	<a href="#">DiagnosticFunctionIdentifier</a>	0..1	ref	This represents the mapped FID.
mappedSwc Service Dependency	<a href="#">SwcServiceDependency</a>	0..1	iref	This represents the ability to point into the component hierarchy (under possible consideration of the root SoftwareComposition).

**Table 7.12: DiagnosticFimFunctionMapping**

<b>Class</b>	<b>FunctionInhibitionNeeds</b>			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
<b>Note</b>	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.			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a> , <a href="#">ServiceNeeds</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 7.13: FunctionInhibitionNeeds**

## 8 Diagnostics on J1939

### 8.1 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 in a similar way 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.

That said, a support for Dcm-related model elements for J1939 is mainly happening in the domain of the AUTOSAR System Template [6] 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` 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.

## 8.2 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 the 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` it is possible to "inject" the J1939 signal identifier into a `SystemSignal` without actually touching the `SystemSignal`.

**[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*. ] ([RS\\_DEXT\\_00064](#))

Please note that the modeling of the `DiagnosticJ1939SpnMapping` also implies that the same SPN can be sent by different `J1939NmNodes`. This ability is positively supported.

<b>Class</b>	<code>DiagnosticJ1939Spn</code>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::J1939			
<b>Note</b>	This meta-class represents the ability to model a J1939 Suspect Parameter Number (SPN). <b>Tags:</b> atp.recommendedPackage=DiagnosticJ1939Spns			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
spn	PositiveInteger	1	attr	This attribute represents the concrete numerical identification for the enclosing SPN.

**Table 8.1: DiagnosticJ1939Spn**

### 8.3 J1939Dcm-related Modeling

The modeling of the J1939Dcm-related configuration within the Diagnostic Extract extends to the definition of a new attribute of meta-class `J1939Cluster`. The latter provides an attribute named `networkId`. For more information, please refer to [6].

The actual configuration of the services supported by J1939Dcm is done by assigning DMx messages in the System Description.

### 8.4 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”<sup>1</sup> hub for all the Dem-related model elements. The `DiagnosticJ1939Node` represents a specific function, or in terms of the J1939 terminology, a *Controller Application*.

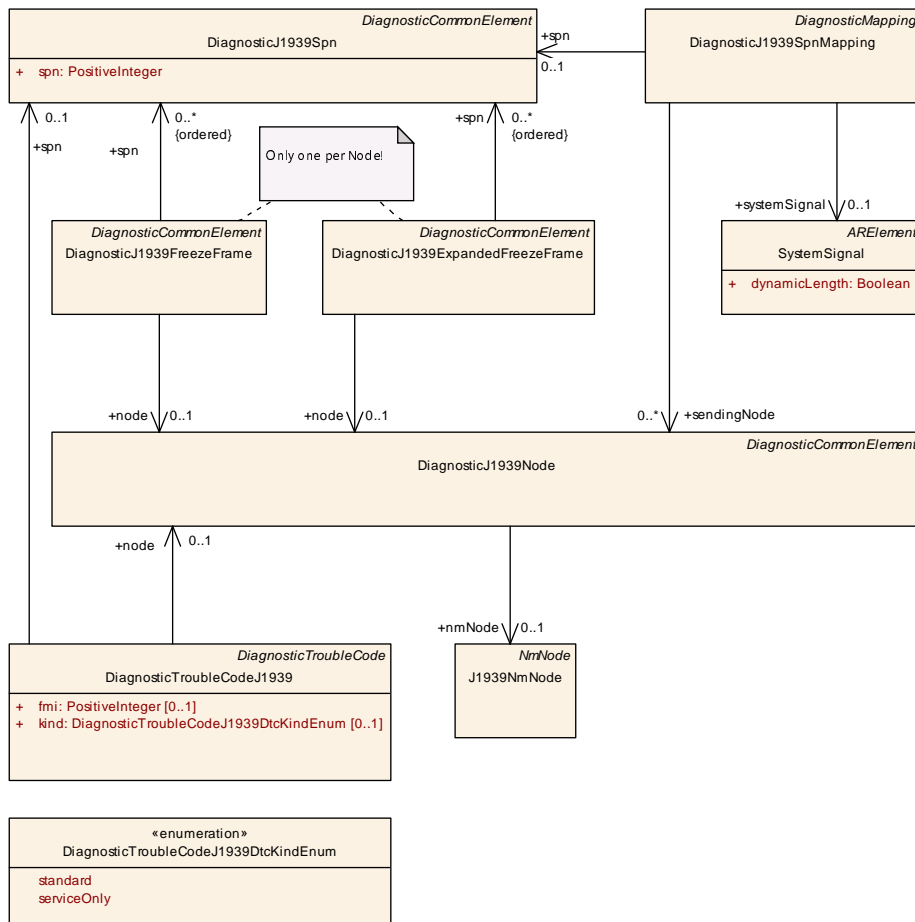
This aspect is stressed in Figure 8.1.

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`** [ Each `J1939NmNode` shall only be referenced in the role `nmNode` by a single `DiagnosticJ1939Node`. ]()

---

<sup>1</sup>Meaning: several meta-classes are referencing the `DiagnosticJ1939Node`.



**Figure 8.1: Big Picture of J1939 diagnostics configuration**

J1939 supports the capturing of information similar to the definition of `DiagnosticFreezeFrame`, as explained in chapter 6.5. 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**. ]([RS\\_DEXT\\_00065](#))

**[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. ]([RS\\_DEXT\\_00065](#))



<b>Class</b>	<b>DiagnosticJ1939FreezeFrame</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::J1939			
<b>Note</b>	This meta-class represents the ability to model a J1939 Freeze Frame. <b>Tags:</b> atp.recommendedPackage=DiagnosticJ1939FreezeFrames			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">Multilanguage</a> , <a href="#">Referrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
node	<a href="#">DiagnosticJ1939Node</a>	0..1	ref	This represents the DiagnosticJ1939Node to which the J1939 freeze frame is associated.
spn (ordered)	<a href="#">DiagnosticJ1939Spn</a>	*	ref	This represents the collection of SPNs that make the J1939 Freeze Frame.

**Table 8.2: DiagnosticJ1939FreezeFrame**

<b>Class</b>	<b>DiagnosticJ1939ExpandedFreezeFrame</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::J1939			
<b>Note</b>	This meta-class represents the ability to model an expanded J1939 Freeze Frame. <b>Tags:</b> atp.recommendedPackage=DiagnosticJ1939ExpandedFreezeFrames			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">Multilanguage</a> , <a href="#">Referrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
node	<a href="#">DiagnosticJ1939Node</a>	0..1	ref	This represents the DiagnosticJ1939Node to which the J1939 expanded freeze frame is associated.
spn (ordered)	<a href="#">DiagnosticJ1939Spn</a>	*	ref	This represents the collection of SPNs that make the expanded J1939 Freeze Frame.

**Table 8.3: DiagnosticJ1939ExpandedFreezeFrame**

<b>Enumeration</b>	<b>DiagnosticTroubleCodeJ1939DtcKindEnum</b>		
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode		
<b>Note</b>	This meta-class represents the ability to further specify a J1939 DTC in terms of its semantics.		
<b>Literal</b>	<b>Description</b>		
serviceOnly	this represents a DTC that is only relevant for service in a garage, reported by e.g. DM53. <b>Tags:</b> atp.EnumerationValue=0		
standard	This represents a non-specific DTC reported by e.g. DM1. <b>Tags:</b> atp.EnumerationValue=1		

**Table 8.4: DiagnosticTroubleCodeJ1939DtcKindEnum**

**[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..\* . ] ([RS\\_DEXT\\_00064](#))

<b>Class</b>	<b>DiagnosticJ1939SpnMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::J1939			
<b>Note</b>	<p>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 changes.</p> <p><b>Tags:</b> atp.recommendedPackage=DiagnosticJ1939SpnMappings</p>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMapping</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
sendingNode	<a href="#">DiagnosticJ1939Node</a>	*	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 though the reference targets the DiagnosticJ1939Node the semantics of the reference is bound to the J1939NmNode that is in turn referenced by the DiagnosticJ1939Node.
spn	<a href="#">DiagnosticJ1939Spn</a>	0..1	ref	This reference goes to the SPN that shall be associated with a SystemSignal.
systemSignal	<a href="#">SystemSignal</a>	0..1	ref	This reference goes to the SystemSignal that shall be associated with an SPN.

**Table 8.5: DiagnosticJ1939SpnMapping**

<b>Class</b>	<b>DiagnosticJ1939Node</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::J1939			
<b>Note</b>	<p>This meta-class represents the diagnostic configuration of a J1939 Nm node, which in turn represents a "virtual Ecu" on the J1939 communication bus.</p> <p><b>Tags:</b> atp.recommendedPackage=DiagnosticJ1939Nodes</p>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
nmNode	<a href="#">J1939NmNode</a>	0..1	ref	This represents the reference to the "virtual Ecu" to which the enclosing DiagnosticJ1939Node is associated.

**Table 8.6: DiagnosticJ1939Node**

[constr\_1456] Valid interval for attribute [DiagnosticTroubleCodeJ1939.fmi](#) [ 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 SAE J1939 [20] specification.

<b>Class</b>	<b>DiagnosticTroubleCodeJ1939</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
<b>Note</b>	<p>This meta-class represents the ability to model specific trouble-code related properties for J1939.</p> <p><b>Tags:</b> atp.recommendedPackage=DiagnosticTroubleCodes</p>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticTroubleCode</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			





Class	DiagnosticTroubleCodeJ1939			
Attribute	Type	Mul.	Kind	Note
dtcProps	<a href="#">DiagnosticTroubleCodeProps</a>	0..1	ref	Defined properties associated with the J1939 DTC.
fmi	PositiveInteger	0..1	attr	This attribute represents the behavior of the Failure Mode Indicator.
kind	<a href="#">DiagnosticTroubleCodeJ1939DtcKindEnum</a>	0..1	attr	This attribute further specifies the DTC in terms of its semantics.
node	<a href="#">DiagnosticJ1939Node</a>	0..1	ref	This represents the related DiagnosticJ1939Node.
spn	<a href="#">DiagnosticJ1939Spn</a>	0..1	ref	This represents the related SPN.

**Table 8.7: DiagnosticTroubleCodeJ1939**

[TPS\_DEXT\_01107] **Definition of service-only DTC** [ It is possible to specify whether given DTCs on J1939 are only relevant for a service session in a garage or workshop. The common property of service-only DTCs is that they shall be located in a memory section that is exclusively used by service-only DTCs. ]([RS\\_DEXT\\_00067](#))

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** [ All [DiagnosticTroubleCodeJ1939](#) with attribute `kind` set to the value `serviceOnly` that reference the same [DiagnosticJ1939Node](#) shall also reference the same [DiagnosticTroubleCodeProps.memoryDestination](#). ]()

[[constr\\_1458](#)] **Reference to [DiagnosticMemoryDestination](#)** [ A [DiagnosticMemoryDestination](#) that is referenced by a [DiagnosticTroubleCodeJ1939.dtcProps.memoryDestination](#) 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`. ]()

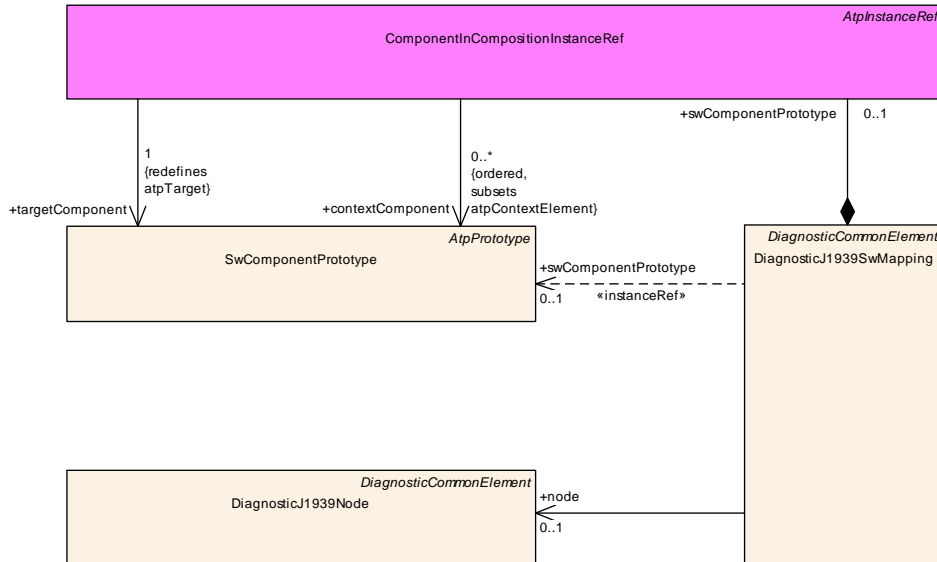
## 8.5 Mapping between Software-Components and Controller Applications

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*. ]([RS\\_DEXT\\_00066](#))

Please note that 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 8.2: Mapping between Software-Components and Controller Applications**

<b>Class</b>	<b>DiagnosticJ1939SwMapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::SwMapping			
<b>Note</b>	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. <b>Tags:</b> atp.recommendedPackage=DiagnosticJ1939SwMappings			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
node	<a href="#">DiagnosticJ1939Node</a>	0..1	ref	This represents the mapped DiagnosticJ1939Node.
swComponentPrototype	<a href="#">SwComponentPrototype</a>	0..1	iref	This represents the mapped SwComponentPrototype.

**Table 8.8: DiagnosticJ1939SwMapping**

## 8.6 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 case of an n:1 mapping, multiple instances of [DiagnosticEventToTroubleCodeJ1939Mapping](#) with the same reference of role `troubleCodeJ1939` but different references of role `diagnosticEvent` have to be defined. | ([RS\\_DEXT\\_00067](#))

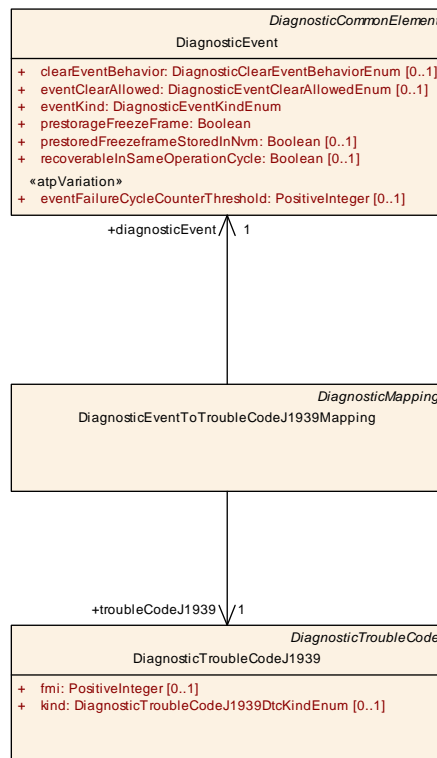


Figure 8.3: Mapping between **DiagnosticEvent** and **DiagnosticTroubleCodeJ1939**

<b>Class</b>	<b>DiagnosticEventToTroubleCodeJ1939Mapping</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::Dem::J1939			
<b>Note</b>	By means of this meta-class it is possible to associate a DiagnosticEvent to a DiagnosticTroubleCode J1939. <b>Tags:</b> atp.recommendedPackage=DiagnosticMappings			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">DiagnosticCommonElement</a> , <a href="#">DiagnosticMapping</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
diagnosticEvent	<a href="#">DiagnosticEvent</a>	1	ref	Reference to a DiagnosticEvent to which a J1939 Diagnostic Trouble Code is assigned.
troubleCode J1939	<a href="#">DiagnosticTroubleCode J1939</a>	1	ref	Reference to a J1939 Diagnostic Trouble Code to which a DiagnosticEvent is assigned.

Table 8.9: DiagnosticEventToTroubleCodeJ1939Mapping

## A 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.

<b>Class</b>	<b>ARElement</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::ARPackage			
<b>Note</b>	An element that can be defined stand-alone, i.e. without being part of another element (except for packages of course).			
<b>Base</b>	<i>ARObject</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
<b>Subclasses</b>	AclObjectSet, AclOperation, AclPermission, AclRole, AliasNameSet, ApplicationPartition, <i>AutosarDataType</i> , <i>BaseType</i> , BlueprintMappingSet, BswEntryRelationshipSet, <i>BswModuleDescription</i> , BswModuleEntry, BuildActionManifest, CalibrationParameterValueSet, ClientIdDefinitionSet, ClientServerInterfaceToBswModuleEntryBlueprintMapping, Collection, <i>CompuMethod</i> , ConsistencyNeedsBlueprintSet, ConstantSpecification, ConstantSpecificationMappingSet, CryptoServiceCertificate, CryptoServiceKey, CryptoServicePrimitive, <i>DataConstr</i> , DataExchangePoint, DataTransformationSet, DataTypeMappingSet, <i>DiagnosticCommonElement</i> , <i>DiagnosticConnection</i> , <i>DiagnosticContributionSet</i> , DiagnosticMasterToSlaveEventMappingSet, Documentation, EcucDefinitionCollection, EcucDestinationUriDefSet, EcucModuleConfigurationValues, EcucModuleDef, EcucValueCollection, EndToEndProtectionSet, EvaluatedVariantSet, FMFeature, FMFeatureMap, FMFeatureModel, FMFeatureSelectionSet, FlatMap, GeneralPurposeConnection, HwCategory, HwElement, HwType, IPv6ExtHeaderFilterSet, <i>Implementation</i> , InterpolationRoutineMappingSet, J1939ControllerApplication, KeywordSet, LifeCycleInfoSet, LifeCycleStateDefinitionGroup, McFunction, McGroup, <i>ModeDeclarationGroup</i> , ModeDeclarationMappingSet, PhysicalDimension, PhysicalDimensionMappingSet, <i>PortInterface</i> , PortInterfaceMappingSet, PortPrototypeBlueprint, PostBuildVariantCriterion, PostBuildVariantCriterionValueSet, PredefinedVariant, RapidPrototypingScenario, SdgDef, SwAddrMethod, SwAxisType, <i>SwComponentType</i> , SwRecordLayout, SwSystemconst, SwSystemconstantValueSet, SwcBswMapping, <i>System</i> , <i>SystemSignal</i> , SystemSignalGroup, TcpOptionFilterSet, <i>TimingExtension</i> , TransformationPropsSet, <i>Unit</i> , UnitGroup, ViewMapSet			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table A.1: ARElement**

<b>Class</b>	<b>ARPackage</b>			
<b>Package</b>	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::ARPackage			
<b>Note</b>	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.			
<b>Base</b>	<i>ARObject</i> , <i>AtpBlueprint</i> , <i>AtpBlueprintable</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
arPackage	<a href="#">ARPackage</a>	*	aggr	This represents a sub package within an ARPackage, thus allowing for an unlimited package hierarchy. <b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30
element	PackageableElement	*	aggr	Elements that are part of this package <b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=20





Class	ARPackage			
referenceBase	ReferenceBase	*	aggr	<p>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.</p> <p><b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=shortLabel xml.sequenceOffset=10</p>

**Table A.2: ARPackage**

Class	<i>AtomicSwComponentType</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::SWComponentTemplate::Components			
<b>Note</b>	An atomic software component is atomic in the sense that it cannot be further decomposed and distributed across multiple ECUs.			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpType</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a> , <a href="#">SwComponentType</a>			
<b>Subclasses</b>	<a href="#">ApplicationSwComponentType</a> , <a href="#">ComplexDeviceDriverSwComponentType</a> , <a href="#">EcuAbstractionSwComponentType</a> , <a href="#">NvBlockSwComponentType</a> , <a href="#">SensorActuatorSwComponentType</a> , <a href="#">ServiceProxySwComponentType</a> , <a href="#">ServiceSwComponentType</a>			
Attribute	Type	Mul.	Kind	Note
internalBehavior	SwcInternalBehavior	0..1	aggr	<p>The SwcInternalBehaviors owned by an AtomicSwComponentType can be located in a different physical file. Therefore the aggregation is «atpSplitable».</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=internalBehavior, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
symbolProps	SymbolProps	0..1	aggr	<p>This represents the SymbolProps for the AtomicSwComponentType.</p> <p><b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=shortName</p>

**Table A.3: AtomicSwComponentType**

Class	<i>AtpInstanceRef</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::GenericStructure::AbstractStructure			
<b>Note</b>	<p>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.</p> <p>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).</p>			
<b>Base</b>	<a href="#">ARObject</a>			
<b>Subclasses</b>	<a href="#">AnyInstanceRef</a> , <a href="#">ApplicationCompositeElementInPortInterfaceInstanceRef</a> , <a href="#">ApplicationDataPrototypeInSystemInstanceRef</a> , <a href="#">ComponentInCompositionInstanceRef</a> , <a href="#">ComponentInSystemInstanceRef</a> , <a href="#">DataPrototypeInSystemInstanceRef</a> , <a href="#">InnerDataPrototypeGroupInCompositionInstanceRef</a> , <a href="#">InnerPortGroupInCompositionInstanceRef</a> , <a href="#">InnerRunnableEntityGroupInCompositionInstanceRef</a> , <a href="#">InstanceEventInCompositionInstanceRef</a> , <a href="#">ModeGroupInAtomicSwcInstanceRef</a> , <a href="#">ModelInBswModuleDescriptionInstanceRef</a> , <a href="#">ModelInSwcInstanceRef</a> , <a href="#">OperationArgumentInComponentInstanceRef</a> , <a href="#">OperationInAtomicSwcInstanceRef</a> , <a href="#">OperationInSystemInstanceRef</a> , <a href="#">PModelInSystemInstanceRef</a> , <a href="#">ParameterInAtomicSWCTypeInstanceRef</a> , <a href="#">PortGroupInSystemInstanceRef</a> , <a href="#">PortInCompositionTypeInstanceRef</a> , <a href="#">RModelInAtomic</a>			







<b>Class</b>	<b>AtpInstanceRef</b> (abstract)			
	<p style="text-align: center;">△</p> <p>SwcInstanceRef, RteEventInEcuInstanceRef, RunnableEntityInCompositionInstanceRef, <a href="#">SwcServiceDependencyInSystemInstanceRef</a>, <a href="#">TriggerInAtomicSwcInstanceRef</a>, TriggerInSystemInstanceRef, VariableAccessInEcuInstanceRef, VariableDataPrototypeInCompositionInstanceRef, VariableDataPrototypeInSystemInstanceRef, VariableInAtomicSWCTypeInstanceRef, <a href="#">VariableInAtomicSwcInstanceRef</a>, VariableInComponentInstanceRef</p>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
atpBase	AtpClassifier	1	ref	This is the base from which the navigation path starts. <b>Stereotypes:</b> atpAbstract; atpDerived
atpContextElement (ordered)	AtpPrototype	*	ref	This is one particular step in the navigation path. <b>Stereotypes:</b> atpAbstract
atpTarget	AtpFeature	1	ref	This is the target of the instance ref. In other words it is the terminal of the navigation path. <b>Stereotypes:</b> atpAbstract

**Table A.4: AtpInstanceRef**

<b>Class</b>	<b>BaseType</b> (abstract)			
<b>Package</b>	M2::MSR::AsamHdo::BaseTypes			
<b>Note</b>	This abstract meta-class represents the ability to specify a platform dependant base type.			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">SwBaseType</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
baseTypeDefinition	BaseTypeDefinition	1	aggr	This is the actual definition of the base type. <b>Tags:</b> xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

**Table A.5: BaseType**

<b>Class</b>	<b>BaseTypeDirectDefinition</b>			
<b>Package</b>	M2::MSR::AsamHdo::BaseTypes			
<b>Note</b>	This BaseType is defined directly (as opposite to a derived BaseType)			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">BaseTypeDefinition</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
baseTypeEncoding	BaseTypeEncodingString	1	attr	This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. <b>Tags:</b> xml.sequenceOffset=90
baseTypeSize	PositiveInteger	0..1	attr	Describes the length of the data type specified in the container in bits. <b>Tags:</b> xml.sequenceOffset=70
byteOrder	ByteOrderEnum	0..1	attr	This attribute specifies the byte order of the base type. <b>Tags:</b> xml.sequenceOffset=110







Class	BaseTypeDirectDefinition			
memAlignment	PositiveInteger	0..1	attr	<p>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".</p> <p><b>Tags:</b> xml.sequenceOffset=100</p>
native Declaration	NativeDeclarationString	0..1	attr	<p>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</p> <p>BaseType with</p> <pre>shortName: "MyUnsignedInt" nativeDeclaration: "unsigned short"</pre> <p>Results in</p> <pre>typedef unsigned short MyUnsignedInt;</pre> <p>If the attribute is not defined the referring Implementation DataTypes will not be generated as a typedef by RTE.</p> <p>If a nativeDeclaration type is given it shall fulfill the characteristic given by basetypeEncoding and baseType Size.</p> <p>This is required to ensure the consistent handling and interpretation by software components, RTE, COM and MCM systems.</p> <p><b>Tags:</b> xml.sequenceOffset=120</p>

**Table A.6: BaseTypeDirectDefinition**

Class	BswModuleDescription			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswOverview			
Note	<p>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.</p> <p><b>Tags:</b> atp.recommendedPackage=BswModuleDescriptions</p>			
Base	<p><a href="#">ARElement</a>, <a href="#">ARObject</a>, <a href="#">AtpBlueprint</a>, <a href="#">AtpBlueprintable</a>, <a href="#">AtpClassifier</a>, <a href="#">AtpFeature</a>, <a href="#">AtpStructureElement</a>, <a href="#">CollectableElement</a>, <a href="#">Identifiable</a>, <a href="#">MultilanguageReferrable</a>, <a href="#">PackageableElement</a>, <a href="#">Referrable</a></p>			
Attribute	Type	Mul.	Kind	Note
bswModule Dependency	BswModuleDependency	*	aggr	<p>Describes the dependency to another BSW module.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=20</p>
bswModule Documentation	SwComponent Documentation	0..1	aggr	<p>This adds a documentation to the BSW module.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=bswModuleDocumentation, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=6</p>





Class	BswModuleDescription			
expectedEntry	BswModuleEntry	*	ref	<p>Indicates an entry which is required by this module. Replacement of outgoingCallback / requiredEntry.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation  <b>Tags:</b> atp.Splitkey=expectedEntry, variationPoint.shortLabel            vh.latestBindingTime=preCompileTime</p>
implementedEntry	BswModuleEntry	*	ref	<p>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.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation  <b>Tags:</b> atp.Splitkey=implementedEntry, variationPoint.shortLabel            vh.latestBindingTime=preCompileTime</p>
internalBehavior	BswInternalBehavior	*	aggr	<p>The various BswInternalBehaviors associated with a BswModuleDescription can be distributed over several physical files. Therefore the aggregation is «atpSplitable».</p> <p><b>Stereotypes:</b> atpSplitable  <b>Tags:</b> atp.Splitkey=shortName            xml.sequenceOffset=65</p>
moduleId	PositiveInteger	0..1	attr	<p>Refers to the BSW Module Identifier defined by the AUTOSAR standard. For non-standardized modules, a proprietary identifier can be optionally chosen.</p> <p><b>Tags:</b> xml.sequenceOffset=5</p>
providedClientServerEntry	BswModuleClientServerEntry	*	aggr	<p>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.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation  <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel            vh.latestBindingTime=preCompileTime            xml.sequenceOffset=45</p>
providedData	<a href="#">VariableDataPrototype</a>	*	aggr	<p>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.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation  <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel            vh.latestBindingTime=preCompileTime            xml.sequenceOffset=55</p>
providedModeGroup	<a href="#">ModeDeclarationGroupPrototype</a>	*	aggr	<p>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, EcuAbstractionSwComponentType or ComplexDeviceDriverSwComponentType.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation  <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel            vh.latestBindingTime=preCompileTime            xml.sequenceOffset=25</p>





Class	BswModuleDescription			
releasedTrigger	Trigger	*	aggr	<p>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.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation  <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel            vh.latestBindingTime=preCompileTime            xml.sequenceOffset=35</p>
requiredClientServerEntry	BswModuleClientServerEntry	*	aggr	<p>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.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation  <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel            vh.latestBindingTime=preCompileTime            xml.sequenceOffset=50</p>
requiredData	<a href="#">VariableDataPrototype</a>	*	aggr	<p>Specifies a data prototype required by this module in order 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.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation  <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel            vh.latestBindingTime=preCompileTime            xml.sequenceOffset=60</p>
requiredModeGroup	<a href="#">ModeDeclarationGroup Prototype</a>	*	aggr	<p>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.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation  <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel            vh.latestBindingTime=preCompileTime            xml.sequenceOffset=30</p>
requiredTrigger	Trigger	*	aggr	<p>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.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation  <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel            vh.latestBindingTime=preCompileTime            xml.sequenceOffset=40</p>

**Table A.7: BswModuleDescription**

<b>Class</b>	<b>CompositionSwComponentType</b>			
<b>Package</b>	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
<b>Note</b>	<p>A CompositionSwComponentType aggregates SwComponentPrototypes (that in turn are typed by SwComponentTypes) as well as SwConnectors for primarily connecting SwComponentPrototypes among each others and towards the surface of the CompositionSwComponentType. By this means hierarchical structures of software-components can be created.</p> <p><b>Tags:</b> atp.recommendedPackage=SwComponentTypes</p>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpType</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a> , <a href="#">SwComponentType</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
component	<a href="#">SwComponentPrototype</a>	*	aggr	<p>The instantiated components that are part of this composition.</p> <p>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.</p> <p>The aggregation is marked as atpSplitable in order to allow the addition of service components to the ECU extract during the ECU integration.</p> <p>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.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild</p>
connector	SwConnector	*	aggr	<p>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.</p> <p>The aggregation of SwConnectors is subject to variability with the purpose to support variant data flow.</p> <p>The aggregation is marked as atpSplitable in order to allow the extension of the ECU extract with AssemblySwConnectors between ApplicationSwComponentTypes and ServiceSwComponentTypes during the ECU integration.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild</p>
constantValue Mapping	ConstantSpecification MappingSet	*	ref	<p>Reference to the ConstantSpecificationMapping to be applied for initValues of PPortComSpecs and RPortComSpec.</p> <p><b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=constantValueMapping</p>
dataType Mapping	DataTypeMappingSet	*	ref	<p>Reference to the DataTypeMapping to be applied for the used ApplicationDataTypes in PortInterfaces.</p> <p>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</p>





Class	CompositionSwComponentType			
				<p>△</p> <p>the implementers mainly because the RTE generator is not concerned about the CompositionSwComponent Types.</p> <p>Rationale: if the mapping of ApplicationDataTypes on the delegated and inner PortPrototype matches then the mapping to ImplementationDataTypes is not impacting compatibility.</p> <p><b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=dataTypeMapping</p>
instantiation RTEEventProps	InstantiationRTEEvent Props	*	aggr	<p>This allows to define instantiation specific properties for RTE Events, in particular for instance specific scheduling.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=shortLabel, variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime</p>

**Table A.8: CompositionSwComponentType**

Class	CompuMethod			
<b>Package</b>	M2::MSR::AsamHdo::ComputationMethod			
<b>Note</b>	<p>This meta-class represents the ability to express the relationship between a physical value and the mathematical representation.</p> <p>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.</p> <p><b>Tags:</b> atp.recommendedPackage=CompuMethods</p>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">Multilanguage</a> , <a href="#">Referrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
Attribute	Type	Mul.	Kind	Note
compuInternal ToPhys	Compu	0..1	aggr	<p>This specifies the computation from internal values to physical values.</p> <p><b>Tags:</b> xml.sequenceOffset=80</p>
compuPhysTo Internal	Compu	0..1	aggr	<p>This represents the computation from physical values to the internal values.</p> <p><b>Tags:</b> xml.sequenceOffset=90</p>
displayFormat	DisplayFormatString	0..1	attr	<p>This property specifies, how the physical value shall be displayed e.g. in documents or measurement and calibration tools.</p> <p><b>Tags:</b> xml.sequenceOffset=20</p>
unit	<a href="#">Unit</a>	0..1	ref	<p>This is the physical unit of the Physical values for which the CompuMethod applies.</p> <p><b>Tags:</b> xml.sequenceOffset=30</p>

**Table A.9: CompuMethod**

<b>Class</b>	<b>DataConstr</b>			
<b>Package</b>	M2::MSR::AsamHdo::Constraints::GlobalConstraints			
<b>Note</b>	This meta-class represents the ability to specify constraints on data. <b>Tags:</b> atp.recommendedPackage=DataConstrs			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataConstrRule	DataConstrRule	*	aggr	This is one particular rule within the data constraints. <b>Tags:</b> xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=30 xml.typeElement=false xml.typeWrapperElement=false

**Table A.10: DataConstr**

<b>Class</b>	<b>DataInterface</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
<b>Note</b>	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).			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpType</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">PortInterface</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	NvDataInterface, ParameterInterface, <a href="#">SenderReceiverInterface</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table A.11: DataInterface**

<b>Class</b>	<b>DataPrototype</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
<b>Note</b>	Base class for prototypical roles of any data type.			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">AtpFeature</a> , <a href="#">AtpPrototype</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">ApplicationCompositeElementDataPrototype</a> , <a href="#">AutosarDataPrototype</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
swDataDef Props	<a href="#">SwDataDefProps</a>	0..1	aggr	This property allows to specify data definition properties which apply on data prototype level.

**Table A.12: DataPrototype**

<b>Class</b>	<b>EcuInstance</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
<b>Note</b>	<p>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.</p> <p><b>Tags:</b> atp.recommendedPackage=EcuInstances</p>			
<b>Base</b>	<i>ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
associatedCommunicationPduGroup	<a href="#">ISignalPduGroup</a>	*	ref	<p>With this reference it is possible to identify which ISignalPduGroups are applicable for which Communication Connector/ ECU.</p> <p>Only top level ISignalPduGroups shall be referenced by an EcuInstance. If an ISignalPduGroup contains other ISignalPduGroups than these contained ISignalPduGroups shall not be referenced by the EcuInstance. Contained ISignalPduGroups are associated to an Ecu Instance via the top level ISignalPduGroup.</p>





Class	EcuInstance			
associatedPdur IPduGroup	PdurIPduGroup	*	ref	With this reference it is possible to identify which PduR IPdu Groups are applicable for which Communication Connector/ ECU.
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.
com Configuration GwTimeBase	TimeValue	0..1	attr	The period between successive calls to Com_Main FunctionRouteSignals of the AUTOSAR COM module in seconds.
com ConfigurationRx TimeBase	TimeValue	0..1	attr	The period between successive calls to Com_Main FunctionRx of the AUTOSAR COM module in seconds.
com ConfigurationTx TimeBase	TimeValue	0..1	attr	The period between successive calls to Com_Main FunctionTx of the AUTOSAR COM module in seconds.
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 numberOfRepetitions > 0).
commController	Communication Controller	1..*	aggr	CommunicationControllers of the ECU.
connector	Communication Connector	*	aggr	All channels controlled by a single controller.
diagnostic Address	Integer	0..1	attr	An ECU specific ID for responses of diagnostic routines.
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.
partition	EcuPartition	*	aggr	Optional definition of Partitions within an Ecu.
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.
pncPrepare SleepTimer	TimeValue	0..1	attr	Time in seconds the PNC state machine shall wait in PNC_PREPARE_SLEEP.
sleepMode Supported	Boolean	1	attr	Specifies whether the ECU instance may be put to a "low power mode" <ul style="list-style-type: none"> <li>• true: sleep mode is supported</li> <li>• false: sleep mode is not supported</li> </ul> <p>Note: This flag may only be set to "true" if the feature is supported by both hardware and basic software.</p>
v2xSupported	V2xSupportEnum	0..1	attr	This attribute is used to control the existence of the V2X stack on the given EcuInstance.
wakeUpOver BusSupported	Boolean	1	attr	Driver support for wakeup over Bus.

**Table A.13: EcuInstance**



<b>Class</b>	<b>ISignal</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
<b>Note</b>	<p>Signal of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal is sent in different SignallPdu to multiple receivers.</p> <p>To support the RTE "signal fan-out" each SignallPdu contains ISignals. If the same System Signal is to be mapped into several SignallPdu there is one ISignal needed for each ISignalToIPduMapping.</p> <p>ISignals describe the Interface between the Precompile configured RTE and the potentially Postbuild configured Com Stack (see ECUC Parameter Mapping).</p> <p>In case of the SystemSignalGroup an ISignal must be created for each SystemSignal contained in the SystemSignalGroup.</p> <p><b>Tags:</b> atp.recommendedPackage=ISignals</p>			
<b>Base</b>	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataTransformation	DataTransformation	0..1	ref	<p>Optional reference to a DataTransformation which represents the transformer chain that is used to transform the data that shall be placed inside this ISignal.</p> <p><b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=dataTransformation, variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime</p>
dataTypePolicy	DataTypePolicyEnum	1	attr	<p>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.</p> <p>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.</p> <p>In case the System Description doesn't use a complete Software Component Description (VFB View) the "legacy" policy can be chosen.</p>
iSignalProps	ISignalProps	0..1	aggr	<p>Additional optional ISignal properties that may be stored in different files.</p> <p><b>Stereotypes:</b> atpSplittable <b>Tags:</b> atp.Splitkey=iSignalProps</p>
iSignalType	ISignalTypeEnum	0..1	attr	<p>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.</p>
initValue	ValueSpecification	0..1	aggr	<p>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.</p> <p>This value can be used to configure the Signal's "Init Value".</p> <p>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.</p>





Class	ISignal			
length	Integer	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.
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.
systemSignal	SystemSignal	1	ref	Reference to the System Signal that is supposed to be transmitted in the ISignal.
timeout Substitution Value	ValueSpecification	0..1	aggr	Defines and enables the ComTimeoutSubstitution for this ISignal.
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.

**Table A.14: ISignal**

Class	ISignalIPduGroup			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
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. <b>Tags:</b> atp.recommendedPackage=ISignaliPduGroup			
Base	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable			
Attribute	Type	Mul.	Kind	Note
communication Direction	Communication DirectionType	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	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.





Class	ISignalPduGroup			
contained ISignalPdu Group	<a href="#">ISignalPduGroup</a>	*	ref	An I-Pdu group can be included in other I-Pdu groups. Contained I-Pdu groups shall not be referenced by the EcuInstance.
iSignalPdu	ISignalPdu	*	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).  <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=postBuild
nmPdu	NmPdu	*	ref	Reference to a set of NmPdus with NmUserData, which are contained in the ISignalPduGroup.  <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=postBuild

**Table A.15: ISignalPduGroup**

<b>Class</b>	<i>IdentCaption</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::SWComponentTemplate::RPTScenario			
<b>Note</b>	This meta-class represents the caption. This allows having some meta classes optionally identifiable.			
<b>Base</b>	<i>ARObject</i> , <i>AtpClassifier</i> , <i>AtpFeature</i> , <i>AtpStructureElement</i> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">BswServiceDependencyIdent</a> , <a href="#">ExternalTriggeringPointIdent</a> , <a href="#">ModeAccessPointIdent</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
—	—	—	—	—

**Table A.16: IdentCaption**

<b>Class</b>	<i>Identifiable</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable			
<b>Note</b>	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.			
<b>Base</b>	<i>ARObject</i> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">ARPackage</a> , <a href="#">AbstractEvent</a> , <a href="#">AbstractImplementationDataTypeElement</a> , <a href="#">AbstractServiceInstance</a> , <a href="#">ApplicationEndpoint</a> , <a href="#">ApplicationError</a> , <a href="#">ApplicationPartitionToEcuPartitionMapping</a> , <a href="#">AsynchronousServerCallResultPoint</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpFeature</a> , <a href="#">AutosarOperationArgumentInstance</a> , <a href="#">AutosarVariableInstance</a> , <a href="#">BswInternalTriggeringPoint</a> , <a href="#">BswModuleDependency</a> , <a href="#">BuildActionEntity</a> , <a href="#">BuildActionEnvironment</a> , <a href="#">CanTpAddress</a> , <a href="#">CanTpChannel</a> , <a href="#">CanTpNode</a> , <a href="#">Chapter</a> , <a href="#">ClassContentConditional</a> , <a href="#">ClientIdDefinition</a> , <a href="#">ClientServerOperation</a> , <a href="#">Code</a> , <a href="#">CollectableElement</a> , <a href="#">ComManagementMapping</a> , <a href="#">CommConnectorPort</a> , <a href="#">CommunicationConnector</a> , <a href="#">CommunicationController</a> , <a href="#">Compiler</a> , <a href="#">ConsistencyNeeds</a> , <a href="#">ConsumedEventGroup</a> , <a href="#">CouplingPort</a> , <a href="#">CouplingPortStructuralElement</a> , <a href="#">CryptoServiceMapping</a> , <a href="#">DataPrototypeGroup</a> , <a href="#">DataTransformation</a> , <a href="#">DependencyOnArtifact</a> , <a href="#">DiagEventDebounceAlgorithm</a> , <a href="#">DiagnosticConnectedIndicator</a> , <a href="#">DiagnosticDataElement</a> , <a href="#">DiagnosticFunctionInhibitSource</a> , <a href="#">DiagnosticMasterToSlaveEventMapping</a> , <a href="#">DiagnosticRoutineSubfunction</a> , <a href="#">DolpLogicAddress</a> , <a href="#">ECUMapping</a> , <a href="#">EOCExecutableEntityRefAbstract</a> , <a href="#">EcuPartition</a> , <a href="#">EcucContainerValue</a> , <a href="#">EcucDefinitionElement</a> , <a href="#">EcucDestinationUriDef</a> , <a href="#">EcucEnumerationLiteralDef</a> , <a href="#">EcucQuery</a> , <a href="#">EcucValidationCondition</a> , <a href="#">EndToEndProtection</a> , <a href="#">ExclusiveArea</a> , <a href="#">ExecutableEntity</a> , <a href="#">ExecutionTime</a> , <a href="#">FMAttributeDef</a> , <a href="#">FMFeatureMapAssertion</a> , <a href="#">FMFeatureMapCondition</a> , <a href="#">FMFeatureMapElement</a> , <a href="#">FMFeatureRelation</a> , <a href="#">FMFeatureRestriction</a> , <a href="#">FMFeatureSelection</a> , <a href="#">FlatInstanceDescriptor</a> , <a href="#">FlexrayArTpNode</a> , <a href="#">FlexrayTpConnectionControl</a> , <a href="#">FlexrayTpNode</a> , <a href="#">FlexrayTpPduPool</a> , <a href="#">FrameTriggering</a> , <a href="#">GeneralParameter</a> , <a href="#">GlobalTimeGateway</a> , <a href="#">GlobalTimeMaster</a> , <a href="#">GlobalTimeSlave</a> , <a href="#">HeapUsage</a> , <a href="#">HwAttributeDef</a> , <a href="#">HwAttributeLiteralDef</a> , <a href="#">HwPin</a> , <a href="#">HwPinGroup</a> , <a href="#">IPv6Ext</a>			





Class	Identifiable (abstract)			
	<p>△</p> <p>HeaderFilterList, ISignalToIpduMapping, ISignalTriggering, <a href="#">IdentCaption</a>, InternalTriggeringPoint, J1939SharedAddressCluster, J1939TpNode, Keyword, LifeCycleState, LinScheduleTable, LinTpNode, Linker, MacMulticastGroup, McDataInstance, MemorySection, <a href="#">ModeDeclaration</a>, ModeDeclarationMapping, ModeSwitchPoint, NetworkEndpoint, <a href="#">NmCluster</a>, NmEcu, <a href="#">NmNode</a>, NvBlockDescriptor, <a href="#">PackageableElement</a>, ParameterAccess, PduToFrameMapping, PduTriggering, PerInstanceMemory, <a href="#">PhysicalChannel</a>, PortGroup, <a href="#">PortInterfaceMapping</a>, PossibleErrorReaction, ResourceConsumption, RootSwCompositionPrototype, RptComponent, RptContainer, RptExecutableEntity, RptExecutableEntityEvent, RptExecutionContext, RptProfile, RptServicePoint, RunnableEntityGroup, <a href="#">SdgAttribute</a>, SdgClass, SecureCommunicationAuthenticationProps, SecureCommunicationFreshnessProps, <a href="#">ServerCallPoint</a>, <a href="#">ServiceNeeds</a>, SocketAddress, SomeipTpChannel, <a href="#">SpecElementReference</a>, <a href="#">StackUsage</a>, <a href="#">StructuredReq</a>, SwGenericAxisParamType, SwServiceArg, <a href="#">SwcServiceDependency</a>, SwcToApplicationPartitionMapping, SwcToEcuMapping, SwcToImplMapping, SystemMapping, TcpOptionFilterList, TimingCondition, <a href="#">TimingConstraint</a>, <a href="#">TimingDescription</a>, TimingExtensionResource, TimingModelInstance, TlsCryptoCipherSuite, Topic1, TpAddress, <a href="#">TraceableText</a>, <a href="#">TracedFailure</a>, <a href="#">TransformationProps</a>, TransformationTechnology, Trigger, VariableAccess, VariationPointProxy, ViewMap, VlanConfig, WaitPoint</p>			
Attribute	Type	Mul.	Kind	Note
desc	<a href="#">MultiLanguageOverviewParagraph</a>	0..1	aggr	<p>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.</p> <p>More elaborate documentation, (in particular how the object is built or used) should go to "introduction".</p> <p><b>Tags:</b> xml.sequenceOffset=-60</p>
category	CategoryString	0..1	attr	<p>The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.</p> <p><b>Tags:</b> xml.sequenceOffset=-50</p>
adminData	AdminData	0..1	aggr	<p>This represents the administrative data for the identifiable object.</p> <p><b>Tags:</b> xml.sequenceOffset=-40</p>
annotation	Annotation	*	aggr	<p>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.</p> <p><b>Tags:</b> xml.sequenceOffset=-25</p>
introduction	<a href="#">DocumentationBlock</a>	0..1	aggr	<p>This represents more information about how the object in question is built or is used. Therefore it is a DocumentationBlock.</p> <p><b>Tags:</b> xml.sequenceOffset=-30</p>
uuid	String	0..1	attr	<p>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.</p> <p>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.</p>





<b>Class</b>	<b>Identifiable</b> (abstract)			
				<p style="text-align: center;">△</p> <p>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. <b>Tags:</b> xml.attribute=true</p>

**Table A.17: Identifiable**

<b>Class</b>	<b>InternalBehavior</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::InternalBehavior			
<b>Note</b>	Common base class (abstract) for the internal behavior of both software components and basic software modules/clusters.			
<b>Base</b>	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	BswInternalBehavior, SwcInternalBehavior			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
constantMemory	ParameterDataPrototype	*	aggr	<p>Describes a read only memory object containing characteristic value(s) implemented by this Internal Behavior.</p> <p>The shortName of ParameterDataPrototype has to be equal to the 'C' identifier of the described constant.</p> <p>The characteristic value(s) might be shared between SwComponentPrototypes of the same SwComponent Type.</p> <p>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.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel.vh.latestBindingTime=preCompileTime</p>
constantValueMapping	ConstantSpecificationMappingSet	*	ref	<p>Reference to the ConstanSpecificationMapping to be applied for the particular InternalBehavior</p> <p><b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=constantValueMapping</p>
dataTypeMapping	DataTypeMappingSet	*	ref	<p>Reference to the DataTypeMapping to be applied for the particular InternalBehavior</p> <p><b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=dataTypeMapping</p>
exclusiveArea	ExclusiveArea	*	aggr	<p>This specifies an ExclusiveArea for this InternalBehavior. The exclusiveArea is local to the component resp. module.</p> <p>The aggregation of ExclusiveAreas is subject to variability. Note: the number of ExclusiveAreas might vary due to the</p> <p style="text-align: center;">▽</p>





Class	InternalBehavior (abstract)			
				<p>△</p> <p>conditional existence of RunnableEntities or BswModule Entities.</p> <p><b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
exclusiveAreaNestingOrder	ExclusiveAreaNestingOrder	*	aggr	<p>This represents the set of ExclusiveAreaNestingOrder owned by the InternalBehavior.</p> <p><b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>
staticMemory	<a href="#">VariableDataPrototype</a>	*	aggr	<p>Describes a read and writeable static memory object representing measurement variables implemented by this software component.</p> <p>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.</p> <p>The shortName of the VariableDataPrototype has to be equal with the "C" identifier of the described variable.</p> <p>The aggregation of staticMemory is subject to variability with the purpose to support variability in the software component's implementations.</p> <p>Typically different algorithms in the implementation are requiring different number of memory objects.</p> <p><b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p>

**Table A.18: InternalBehavior**

<b>Class</b>	«atpVariation» <b>J1939Cluster</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
<b>Note</b>	<p>J1939 specific cluster attributes.</p> <p><b>Tags:</b> atp.recommendedPackage=CommunicationClusters</p>			
<b>Base</b>	<p><i>ARObject</i>, <i>AbstractCanCluster</i>, <i>CollectableElement</i>, <i>CommunicationCluster</i>, <i>FibexElement</i>, <i>Identifiable</i>, <i>MultilanguageReferrable</i>, <i>PackageableElement</i>, <i>Referrable</i></p>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
networkId	PositiveInteger	0..1	attr	This represents the network ID for the J1939 cluster.
request2Support	Boolean	0..1	attr	Enables support for the Request2 PGN (RQST2).
usesAddressArbitration	Boolean	0..1	attr	<p>Defines whether the nodes attached to this channel use an initial address claim, and whether they react to contending address claims of other nodes.</p> <p>True: The initial address claim is sent, and the node reacts to address claims of other nodes.</p> <p>False: The node only sends an address claim upon request, and does not care for contending address claims.</p>

**Table A.19: J1939Cluster**

<b>Class</b>	<b>J1939DcmIPdu</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
<b>Note</b>	Represents the IPdus handled by J1939Dcm. <b>Tags:</b> atp.recommendedPackage=Pdus			
<b>Base</b>	ARObject, CollectableElement, FibexElement, IPdu, <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Pdu</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
diagnosticMessageType	PositiveInteger	0..1	attr	This attribute is used to identify the actual DMx message, e.g 1 means DM01, etc.

**Table A.20: J1939DcmIPdu**

<b>Class</b>	<b>J1939NmNode</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
<b>Note</b>	J1939 specific NM Node attributes.			
<b>Base</b>	ARObject, <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">NmNode</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
nodeName	J1939NodeName	0..1	aggr	nodeName configuration

**Table A.21: J1939NmNode**

<b>Class</b>	<b>ModeDeclaration</b>			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
<b>Note</b>	Declaration of one Mode. The name and semantics of a specific mode is not defined in the meta-model. <b>Tags:</b> atp.ManifestKind=ExecutionManifest,MachineManifest			
<b>Base</b>	ARObject, <a href="#">AtpClassifier</a> , <a href="#">AtpFeature</a> , <a href="#">AtpStructureElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
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 A.22: ModeDeclaration**

<b>Class</b>	<b>ModeDeclarationGroup</b>			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
<b>Note</b>	A collection of Mode Declarations. Also, the initial mode is explicitly identified. <b>Tags:</b> atp.ManifestKind=ExecutionManifest,MachineManifest atp.recommendedPackage=ModeDeclarationGroups			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpType</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
initialMode	<a href="#">ModeDeclaration</a>	1	ref	The initial mode of the ModeDeclarationGroup. This mode is active before any mode switches occurred.
modeDeclaration	<a href="#">ModeDeclaration</a>	1..*	aggr	The ModeDeclarations collected in this ModeDeclaration Group. <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=blueprintDerivationTime







Class	ModeDeclarationGroup			
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).
modeTransition	ModeTransition	*	aggr	This represents the available ModeTransitions of the ModeDeclarationGroup
modeUserError Behavior	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).
onTransition Value	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.

**Table A.23: ModeDeclarationGroup**

Class	ModeDeclarationGroupPrototype			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	The ModeDeclarationGroupPrototype specifies a set of Modes (ModeDeclarationGroup) which is provided or required in the given context. <b>Tags:</b> atp.ManifestKind=ExecutionManifest,MachineManifest			
Base	ARObject, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, Referrable			
Attribute	Type	Mul.	Kind	Note
swCalibration Access	SwCalibrationAccess Enum	0..1	attr	This allows for specifying whether or not the enclosing ModeDeclarationGroupPrototype can be measured at run-time.
type	ModeDeclarationGroup	1	tref	The "collection of ModeDeclarations" (= ModeDeclarationGroup) supported by a component <b>Stereotypes:</b> isOfType

**Table A.24: ModeDeclarationGroupPrototype**

Class	ModelnBswModuleDescriptionInstanceRef			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswOverview::InstanceRefs			
Note				
Base	ARObject, AtpInstanceRef			
Attribute	Type	Mul.	Kind	Note
base	BswModuleDescription	1	ref	<b>Stereotypes:</b> atpDerived <b>Tags:</b> xml.sequenceOffset=10
contextMode Declaration Group	ModeDeclarationGroup Prototype	1	ref	<b>Tags:</b> xml.sequenceOffset=20
targetMode	ModeDeclaration	1	ref	<b>Tags:</b> xml.sequenceOffset=30

**Table A.25: ModelnBswModuleDescriptionInstanceRef**



<b>Primitive</b>	<b>NameToken</b>
<b>Package</b>	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes
<b>Note</b>	This is an identifier as used in xml, e.g. xml-names. Basic difference to Identifier is the fact that it can contain "-".  <b>Tags:</b> xml.xsd.customType=NMOKEN-STRING xml.xsd.type=NMOKEN

**Table A.26: NameToken**

<b>Class</b>	<b>PPortPrototype</b>			
<b>Package</b>	M2::AUTOSARTemplates::SWComponentTemplate::Components			
<b>Note</b>	Component port providing a certain port interface.			
<b>Base</b>	<i>ARObject</i> , <i>AbstractProvidedPortPrototype</i> , <i>AtpBlueprintable</i> , <i>AtpFeature</i> , <i>AtpPrototype</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PortPrototype</i> , <i>Referrable</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
provided Interface	<a href="#">PortInterface</a>	1	tref	The interface that this port provides.  <b>Stereotypes:</b> isOfType

**Table A.27: PPortPrototype**

<b>Class</b>	<b>PortInterface</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
<b>Note</b>	Abstract base class for an interface that is either provided or required by a port of a software component.			
<b>Base</b>	<i>ARElement</i> , <i>ARObject</i> , <i>AtpBlueprint</i> , <i>AtpBlueprintable</i> , <i>AtpClassifier</i> , <i>AtpType</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
<b>Subclasses</b>	<i>ClientServerInterface</i> , <i>DataInterface</i> , <i>ModeSwitchInterface</i> , <i>TriggerInterface</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
isService	Boolean	1	attr	This flag is set if the PortInterface is to be used for communication between an <ul style="list-style-type: none"> <li>• ApplicationSwComponentType or</li> <li>• ServiceProxySwComponentType or</li> <li>• SensorActuatorSwComponentType or</li> <li>• ComplexDeviceDriverSwComponentType</li> <li>• ServiceSwComponentType</li> <li>• EcuAbstractionSwComponentType</li> </ul> and a ServiceSwComponentType (namely an AUTOSAR Service) located on the same ECU. Otherwise the flag is not set.
serviceKind	ServiceProviderEnum	0..1	attr	This attribute provides further details about the nature of the applied service.

**Table A.28: PortInterface**

<b>Class</b>	<b>PortPrototype</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::SWComponentTemplate::Components			
<b>Note</b>	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.			
<b>Base</b>	ARObject, AtpBlueprintable, AtpFeature, AtpPrototype, <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	AbstractProvidedPortPrototype, AbstractRequiredPortPrototype			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
clientServer Annotation	ClientServerAnnotation	*	aggr	Annotation of this PortPrototype with respect to client/server communication.
delegatedPort Annotation	DelegatedPort Annotation	0..1	aggr	Annotations on this delegated port.
ioHwAbstractionServer Annotation	IoHwAbstractionServer Annotation	*	aggr	Annotations on this IO Hardware Abstraction port.
modePort Annotation	ModePortAnnotation	*	aggr	Annotations on this mode port.
nvDataPort Annotation	NvDataPortAnnotation	*	aggr	Annotations on this non volatile data port.
parameterPort Annotation	ParameterPort Annotation	*	aggr	Annotations on this parameter port.
senderReceiver Annotation	SenderReceiver Annotation	*	aggr	Collection of annotations of this ports sender/receiver communication.
triggerPort Annotation	TriggerPortAnnotation	*	aggr	Annotations on this trigger port.

**Table A.29: PortPrototype**

<b>Class</b>	<b>Referrable</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable			
<b>Note</b>	Instances of this class can be referred to by their identifier (while adhering to namespace borders).			
<b>Base</b>	ARObject			
<b>Subclasses</b>	AtpDefinition, BswDistinguishedPartition, BswModuleCallPoint, BswModuleClientServerEntry, BswVariableAccess, CouplingPortTrafficClassAssignment, <a href="#">DiagnosticDebounceAlgorithmProps</a> , <a href="#">DiagnosticEnvModeElement</a> , EthernetPriorityRegeneration, EventHandler, ExclusiveAreaNestingOrder, HwDescriptionEntity, ImplementationProps, LinSlaveConfigIdent, ModeTransition, <a href="#">MultilanguageReferrable</a> , PncMappingIdent, <a href="#">SingleLanguageReferrable</a> , SocketConnectionBundle, TimeSyncServerConfiguration, TpConnectionIdent			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
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. <b>Tags:</b> xml.enforceMinMultiplicity=true xml.sequenceOffset=-100
shortName Fragment	ShortNameFragment	*	aggr	This specifies how the Referrable.shortName is composed of several shortNameFragments. <b>Tags:</b> xml.sequenceOffset=-90

**Table A.30: Referrable**

<b>Class</b>	<b>SenderReceiverInterface</b>			
<b>Package</b>	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
<b>Note</b>	A sender/receiver interface declares a number of data elements to be sent and received. <b>Tags:</b> atp.recommendedPackage=PortInterfaces			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpType</a> , <a href="#">CollectableElement</a> , <a href="#">DataInterface</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">PortInterface</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dataElement	<a href="#">VariableDataPrototype</a>	1..*	aggr	The data elements of this SenderReceiverInterface.
invalidation Policy	InvalidationPolicy	*	aggr	InvalidationPolicy for a particular dataElement

**Table A.31: SenderReceiverInterface**

<b>Class</b>	<b>ServiceNeeds</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
<b>Note</b>	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.			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	BswMgrNeeds, ComMgrUserNeeds, CryptoServiceJobNeeds, CryptoServiceNeeds, <i>Diagnostic CapabilityElement</i> , DltUserNeeds, <i>DolpServiceNeeds</i> , EcuStateMgrUserNeeds, ErrorTracerNeeds, FunctionInhibitionAvailabilityNeeds, <i>FunctionInhibitionNeeds</i> , GlobalSupervisionNeeds, HardwareTestNeeds, IndicatorStatusNeeds, J1939RmIncomingRequestServiceNeeds, J1939RmOutgoingRequestServiceNeeds, NvBlockNeeds, SecureOnBoardCommunicationNeeds, SupervisedEntityCheckpointNeeds, SupervisedEntityNeeds, SyncTimeBaseMgrUserNeeds, V2xFacUserNeeds, V2xMUserNeeds, VendorSpecificServiceNeeds			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table A.32: ServiceNeeds**

<b>Class</b>	<b>SwBaseType</b>			
<b>Package</b>	M2::MSR::AsamHdo::BaseTypes			
<b>Note</b>	This meta-class represents a base type used within ECU software. <b>Tags:</b> atp.recommendedPackage=BaseTypes			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">BaseType</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table A.33: SwBaseType**

<b>Class</b>	<b>SwComponentPrototype</b>			
<b>Package</b>	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
<b>Note</b>	Role of a software component within a composition.			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">AtpFeature</a> , <a href="#">AtpPrototype</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
type	<a href="#">SwComponentType</a>	1	tref	Type of the instance. <b>Stereotypes:</b> isOfType

**Table A.34: SwComponentPrototype**

<b>Class</b>	<b>SwComponentType</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::SwComponentTemplate::Components			
<b>Note</b>	Base class for AUTOSAR software components.			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpType</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">AtomicSwComponentType</a> , <a href="#">CompositionSwComponentType</a> , <a href="#">ParameterSwComponentType</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
consistency Needs	ConsistencyNeeds	*	aggr	This represents the collection of ConsistencyNeeds owned by the enclosing SwComponentType.  <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
port	<a href="#">PortPrototype</a>	*	aggr	The PortPrototypes through which this SwComponent Type can communicate.  The aggregation of PortPrototype is subject to variability with the purpose to support the conditional existence of PortPrototypes.  <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
portGroup	PortGroup	*	aggr	A port group being part of this component.  <b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=preCompileTime
swComponent Documentation	SwComponent Documentation	0..1	aggr	This adds a documentation to the SwComponentType.  <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=swComponentDocumentation, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=-10
unitGroup	UnitGroup	*	ref	This allows for the specification of which UnitGroups are relevant in the context of referencing SwComponentType.

**Table A.35: SwComponentType**

<b>Class</b>	<b>System</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate			
<b>Note</b>	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.  <b>Tags:</b> atp.recommendedPackage=Systems			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpFeature</a> , <a href="#">AtpStructureElement</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
clientId DefinitionSet	ClientIdDefinitionSet	*	ref	Set of Client Identifiers that are used for inter-ECU client-server communication in the System.
containerIPdu HeaderByte Order	ByteOrderEnum	0..1	attr	Defines the byteOrder of the header in ContainerIPdus.





Class	System			
ecuExtractVersion	RevisionLabelString	0..1	attr	Version number of the Ecu Extract.
fibexElement	FibexElement	*	ref	<p>Reference to ASAM FIBEX elements specifying Communication and Topology.</p> <p>All Fibex Elements used within a System Description shall be referenced from the System Element.</p> <p>atpVariation: In order to describe a product-line, all Fibex Elements can be optional.</p> <p><b>Stereotypes:</b> atpVariation <b>Tags:</b> vh.latestBindingTime=postBuild</p>
j1939SharedAddressCluster	J1939SharedAddressCluster	*	aggr	<p>Collection of J1939Clusters that share a common address space for the routing of messages.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild</p>
mapping	SystemMapping	*	aggr	<p>Aggregation of all mapping aspects (mapping of SW components to ECUs, mapping of data elements to signals, and mapping constraints).</p> <p>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.</p> <p>This element is not required when the System description is used for a network-only use-case.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild</p>
pncVectorLength	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.
rootSoftwareComposition	RootSwCompositionPrototype	0..1	aggr	<p>Aggregation of the root software composition, containing all software components in the System in a hierarchical structure.</p> <p>This element is not required when the System description is used for a network-only use-case.</p> <p>atpVariation: The RootSwCompositionPrototype can vary.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=systemDesignTime</p>
systemDocumentation	Chapter	*	aggr	<p>Possibility to provide additional documentation while defining the System. The System documentation can be composed of several chapters.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=-10</p>
systemVersion	RevisionLabelString	1	attr	Version number of the System Description.

**Table A.36: System**

<b>Class</b>	<b>SystemSignal</b>			
<b>Package</b>	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
<b>Note</b>	<p>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.</p> <p><b>Tags:</b> atp.recommendedPackage=SystemSignals</p>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
dynamicLength	Boolean	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	<a href="#">SwDataDefProps</a>	0..1	aggr	Specification of the physical representation.

**Table A.37: SystemSignal**

<b>Class</b>	<b>Unit</b>			
<b>Package</b>	M2::MSR::AsamHdo::Units			
<b>Note</b>	<p>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.</p> <p>For the calculation from SI-unit to the defined unit the factor (factorSiToUnit ) and the offset (offsetSiToUnit ) are applied as follows:</p> $x \text{ [unit]} := y * \text{[siUnit]} * \text{factorSiToUnit [unit]/[siUnit]} + \text{offsetSiToUnit [unit]}$ <p>For the calculation from a unit to SI-unit the reciprocal of the factor (factorSiToUnit ) and the negation of the offset (offsetSiToUnit ) are applied.</p> $y \text{ [siUnit]} := (x * \text{[unit]} - \text{offsetSiToUnit [unit]}) / (\text{factorSiToUnit [unit]/[siUnit]})$ <p><b>Tags:</b> atp.recommendedPackage=Units</p>			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
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. <b>Tags:</b> 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. <b>Tags:</b> xml.sequenceOffset=30
offsetSiToUnit	Float	0..1	attr	This is the offset for the conversion from and to siUnits. <b>Tags:</b> xml.sequenceOffset=40
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. <b>Tags:</b> xml.sequenceOffset=50

**Table A.38: Unit**

<b>Class</b>	<i>ValueSpecification</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::CommonStructure::Constants			
<b>Note</b>	Base class for expressions leading to a value which can be used to initialize a data object.			
<b>Base</b>	<i>ARObject</i>			
<b>Subclasses</b>	<i>AbstractRuleBasedValueSpecification</i> , <i>ApplicationValueSpecification</i> , <i>CompositeValueSpecification</i> , <i>ConstantReference</i> , <i>NotAvailableValueSpecification</i> , <i>NumericalValueSpecification</i> , <i>ReferenceValueSpecification</i> , <i>TextValueSpecification</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
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 A.39: ValueSpecification**

<b>Class</b>	<b>VariableDataPrototype</b>			
<b>Package</b>	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
<b>Note</b>	A VariableDataPrototype is used to contain values in an ECU application. This means that most likely a VariableDataPrototype allocates "static" memory on the ECU. In some cases optimization strategies might lead to a situation where the memory allocation can be avoided.  In particular, the value of a VariableDataPrototype is likely to change as the ECU on which it is used executes.			
<b>Base</b>	<i>ARObject</i> , <i>AtpFeature</i> , <i>AtpPrototype</i> , <i>AutosarDataPrototype</i> , <i>DataPrototype</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
initValue	<a href="#">ValueSpecification</a>	0..1	aggr	Specifies initial value(s) of the VariableDataPrototype

**Table A.40: VariableDataPrototype**

## B History of Constraints and Specification Items

### B.1 Constraint History of this Document according to AUTOSAR R4.2.1

#### B.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]	<a href="#">DiagnosticContributionSet</a> is the central part of the <a href="#">DiagnosticExtract</a>
[TPS_DEXT_01004]	<a href="#">DiagnosticContributionSet</a> defines the scope of the <a href="#">DiagnosticExtract</a>
[TPS_DEXT_01005]	<a href="#">DiagnosticContributionSet</a> can exist independently







Number	Heading
[TPS_DEXT_01006]	The role of <code>DiagnosticServiceTables</code> in the context of a <code>DiagnosticContributionSet</code>
[TPS_DEXT_01007]	Common properties of a <code>DiagnosticExtract</code>
[TPS_DEXT_01008]	<code>DiagnosticContributionSet</code> 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 <code>DiagnosticSession.id</code>
[TPS_DEXT_01012]	Rationale for the modeling of the multiplicity of <code>DiagnosticAccessPermission.securityLevel</code>
[TPS_DEXT_01013]	Specification of sub-functions by means of attribute <code>DiagnosticServiceInstance.category</code>
[TPS_DEXT_01014]	Possible values of the <code>category</code> attribute for diagnostic services
[TPS_DEXT_01015]	Meaning of attributes of <code>DiagnosticIOControl</code>
[TPS_DEXT_01016]	The capability <code>returnControlToEcu</code>
[TPS_DEXT_01017]	Meaning of <code>DiagnosticIOControl.dataIdentifier</code>
[TPS_DEXT_01018]	<code>InputOutput Control</code> does not define any sub-functions
[TPS_DEXT_01019]	Correspondence of <code>category</code> values to numerical values mentioned in the ISO 14229-1
[TPS_DEXT_01020]	Manufacturer-specific values for sub-functions of service <code>EcuReset</code>
[TPS_DEXT_01021]	Semantics of <code>DiagnosticEcuReset.customSubFunctionNumber</code>
[TPS_DEXT_01022]	<code>ClearDiagnosticInformation</code> does not define any sub-functions
[TPS_DEXT_01023]	<code>WriteMemoryByAddress</code> does not define any sub-functions
[TPS_DEXT_01024]	<code>ReadMemoryByAddress</code> does not define any sub-functions
[TPS_DEXT_01025]	<code>TransferExit</code> does not define any sub-functions
[TPS_DEXT_01026]	<code>DataTransfer</code> does not define any sub-functions
[TPS_DEXT_01027]	<code>RequestDownload</code> does not define any sub-functions
[TPS_DEXT_01028]	<code>RequestUpload</code> does not define any sub-functions
[TPS_DEXT_01029]	Correspondence of <code>category</code> values to numerical values mentioned in the ISO 14229-1
[TPS_DEXT_01030]	Manufacturer-specific values for sub-functions of service <code>CommunicationControl</code>
[TPS_DEXT_01031]	Semantics of <code>DiagnosticComControl.customSubFunctionNumber</code>
[TPS_DEXT_01032]	Impact of the <code>DiagnosticComControlClass</code> on the state management for <code>CommunicationClusters</code>
[TPS_DEXT_01033]	Semantics of triggers in the context of a <code>DiagnosticResponseOnEvent</code>
[TPS_DEXT_01034]	Sub-functions of the service <code>ReadDTCInformation</code>
[TPS_DEXT_01035]	Existence of <code>DiagnosticRoutine.stop</code> and <code>DiagnosticRoutine.requestResult</code>







Number	Heading
[TPS_DEXT_01036]	Work-flow within the execution of the diagnostic service <code>SecurityAccess</code>
[TPS_DEXT_01037]	Semantics of <code>DiagnosticSecurityAccess.requestSeedId</code>
[TPS_DEXT_01038]	Motivation for making the reference <code>DiagnosticSecurityAccess.securityLevel</code> «atpSplitable»
[TPS_DEXT_01039]	Identification of the sub-function of <code>DiagnosticSessionControl</code>
[TPS_DEXT_01040]	Use case where the <code>DiagnosticExtract</code> refers to software-components
[TPS_DEXT_01041]	Semantics of attribute <code>DiagnosticServiceDataMapping.diagnosticDataElement</code>
[TPS_DEXT_01042]	Dem uses <code>DiagnosticServiceDataMapping</code>
[TPS_DEXT_01043]	Purpose of <code>DiagnosticServiceSwMapping</code>
[TPS_DEXT_01044]	<code>BswServiceDependency</code> 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 <code>DiagnosticServiceSwMapping</code> with respect to routine IDs
[TPS_DEXT_01050]	Consistency of <code>DiagnosticServiceSwMapping</code> with respect to data IDs
[TPS_DEXT_01051]	Consistency of <code>DiagnosticServiceSwMapping</code> with respect to data IDs
[TPS_DEXT_01052]	Existence of attribute <code>DiagnosticServiceInstance.accessPermission</code>
[TPS_DEXT_01053]	Existence of <code>DiagnosticSecurityAccess.securityLevel</code>
[TPS_DEXT_01054]	Existence of <code>DiagnosticDataByIdentifier.dataIdentifier</code>
[TPS_DEXT_01055]	Standardized values of <code>DiagnosticContributionSet.category</code>
[TPS_DEXT_01056]	Applicable values for <code>DiagnosticEcuReset.category</code>
[TPS_DEXT_01057]	Allowed values of <code>DiagnosticComControl.category</code>
[TPS_DEXT_01058]	Standardized values for <code>DiagnosticDynamicallyDefineDataIdentifier.category</code>
[TPS_DEXT_01059]	Applicable values for <code>DiagnosticPeriodicRate.category</code>
[TPS_DEXT_01060]	Applicable values for <code>DiagnosticReadDTCInformation.category</code>
[TPS_DEXT_01061]	Supported scenarios for the definition of <i>access permission</i>
[TPS_DEXT_01062]	Existence of <code>DiagnosticServiceClass.accessPermissionValidity</code> in an incomplete model
[TPS_DEXT_01063]	Existence of <code>DiagnosticServiceClass.accessPermissionValidity</code> in a complete model
[TPS_DEXT_01064]	Textually formulated content attached to <code>DiagnosticTroubleCode</code>
[TPS_DEXT_01065]	Different approaches to provide semi-formal textual content attached to a <code>DiagnosticTroubleCode</code>
[TPS_DEXT_01066]	Standardized values of <code>DiagnosticTroubleCode.introduction.trace</code>





Number	Heading
[TPS_DEXT_01067]	Textually formulated content attached to <a href="#">DiagnosticEvent</a>
[TPS_DEXT_01068]	Textual description with respect to the <a href="#">DiagnosticEvent</a>
[TPS_DEXT_01069]	Standardized values of <a href="#">DiagnosticEvent.introduction.structuredReq</a>
[TPS_DEXT_01070]	Description of textually semi-formal formulated pre- and post-conditions for the validity of <a href="#">DiagnosticAccessPermission</a>
[TPS_DEXT_01071]	Standardized values of <a href="#">DiagnosticAccessPermission.introduction.structuredReq</a>
[TPS_DEXT_01072]	Purpose of attribute <a href="#">DiagnosticDataIdentifier.representsVin</a>
[TPS_DEXT_01073]	Diagnostic properties that are specific to an individual <a href="#">EcuInstance</a>
[TPS_DEXT_01074]	Difference between the attributes <a href="#">DiagnosticComControl.specificChannel</a> and <a href="#">DiagnosticComControl.subNodeChannel</a>
[TPS_DEXT_01075]	standardized values for the attribute <a href="#">DiagnosticControlDTCSetting.category</a>
[TPS_DEXT_01076]	Identification of sub-functions of diagnostic service <a href="#">ControlDTCSetting</a>
[TPS_DEXT_01077]	Modeling of <a href="#">DiagnosticRoutine</a>
[TPS_DEXT_01078]	Not possible to use the attribute <a href="#">category</a> for the identification of the sub-function of diagnostic service <a href="#">RoutineControl</a>
[TPS_DEXT_01079]	Modeling of the arguments to a <a href="#">DiagnosticRoutine</a>
[TPS_DEXT_01080]	Diagnostic Routine needs to be started
[TPS_DEXT_01081]	Modeling of <a href="#">DiagnosticSessionControl</a>
[TPS_DEXT_01082]	Existence of <a href="#">DiagnosticSessionControl.diagnosticSession</a>
[TPS_DEXT_01083]	Semantics of a <a href="#">DiagnosticEvent</a>
[TPS_DEXT_01084]	<a href="#">DiagnosticEvent</a> can be connected to one or multiple indicators
[TPS_DEXT_01085]	Semantics of <a href="#">DiagnosticConditionGroups</a>
[TPS_DEXT_01086]	Reference to <a href="#">DiagnosticOperationCycle</a>
[TPS_DEXT_01087]	Semantics of <a href="#">DiagnosticOperationCycle</a>
[TPS_DEXT_01088]	Semantics of <a href="#">DiagnosticRoutine.id</a>
[TPS_DEXT_01089]	Definition of an <i>identifier</i> of a <a href="#">DiagnosticIOControl</a>
[TPS_DEXT_01090]	Diagnostic service <a href="#">RequestFileTransfer</a> does not define any sub-functions
[TPS_DEXT_03000]	ISO 14229-1 reserves values of <a href="#">DiagnosticTroubleCodeGroup.groupNumber</a>
[TPS_DEXT_03001]	Different types of conditions
[TPS_DEXT_03002]	Two kind of mappings
[TPS_DEXT_03003]	Semantics of <a href="#">DiagnosticEventToTroubleCodeUdsMapping</a>
[TPS_DEXT_03004]	<a href="#">DiagnosticEvent</a> and <a href="#">DiagnosticDebounceAlgorithmProps</a>
[TPS_DEXT_03005]	Existence of <a href="#">DiagnosticEventToDebounceAlgorithmMapping</a>
[TPS_DEXT_03006]	Values of the individual <a href="#">DiagnosticStorageConditions</a>





Number	Heading
[TPS_DEXT_03007]	Semantics of <code>DiagnosticEventPortMapping</code>
[TPS_DEXT_03008]	Semantics of <code>DiagnosticExtendedDataRecord</code>
[TPS_DEXT_03009]	Semantics of <code>DiagnosticFreezeFrame</code>
[TPS_DEXT_03010]	Combination of <code>DiagnosticConditions</code> to <code>DiagnosticCondition-Groups</code>
[TPS_DEXT_03011]	Clearing request for a <code>DiagnosticEvent</code>
[TPS_DEXT_03012]	Three kinds of DTCs
[TPS_DEXT_03013]	Common properties of a DTC
[TPS_DEXT_03014]	Semantics of <code>DiagnosticTroubleCodeGroup</code>
[TPS_DEXT_03015]	<code>EnableConditions</code> have to be put into a <code>DiagnosticEnableCondition-Group</code>
[TPS_DEXT_03016]	<code>StorageConditions</code> have to be put into a <code>DiagnosticStorageCondition-Group</code>
[TPS_DEXT_03017]	Semantics of <code>DiagnosticOperationCyclePortMapping</code>
[TPS_DEXT_03018]	Semantics of <code>DiagnosticEnableConditionPortMapping</code>
[TPS_DEXT_03019]	Semantics of <code>DiagnosticStorageConditionPortMapping</code>
[TPS_DEXT_03020]	Semantics of <code>DiagnosticDemProvidedDataMapping</code>
[TPS_DEXT_03021]	Aging
[TPS_DEXT_03022]	Different kinds of <code>DiagnosticIndicators</code>

**Table B.1: Added Specification Items in 4.2.1**

### B.1.2 Added Constraints in R4.2.1

Number	Heading
[constr_1324]	Existence of attribute <code>DiagnosticDataIdentifier.representsVin</code>
[constr_1325]	Allowed attributes of <code>SwDataDefProps</code> for <code>DiagnosticDataElement.sw-DataDefProps</code>
[constr_1326]	Existence of a variable-sized array
[constr_1327]	Multiplicity of <code>DiagnosticContributionSet.ecuInstance</code>
[constr_1328]	Consistency of <code>DiagnosticContributionSet.ecuInstance</code> and <code>Diagnostic-ServiceTable.ecuInstance</code>
[constr_1329]	Existence of concrete sub-classes of <code>DiagnosticServiceClass</code> in the context created by a <code>DiagnosticContributionSet</code>
[constr_1330]	Custom service identifier shall not overlap with standardized service identifiers
[constr_1331]	Existence of <code>DiagnosticEcuReset.customSubFunctionNumber</code>
[constr_1332]	Value range for <code>DiagnosticEcuReset.customSubFunctionNumber</code>
[constr_1333]	Existence of <code>DiagnosticMemoryIdentifier.memoryLowAddress</code> and <code>DiagnosticMemoryIdentifier.memoryHighAddress</code>





Number	Heading
[constr_1334]	Existence of <code>DiagnosticComControl.customSubFunctionNumber</code>
[constr_1335]	Possible values for <code>DiagnosticComControl.customSubFunctionNumber</code>
[constr_1336]	Applicable value range for <code>DiagnosticComControlSpecificChannel.subnetNumber</code>
[constr_1337]	Allowed value range for attribute <code>DiagnosticComControlSubNodeChannel.subNodeNumber</code>
[constr_1338]	Maximum number of aggregated <code>DiagnosticReadDataByPeriodicIDClass.periodicRate</code>
[constr_1339]	Existence of <code>DiagnosticRoutine.start</code>
[constr_1340]	Consistency of <code>DiagnosticServiceSwMapping</code> with respect to synchronously called <code>DiagnosticRoutines</code>
[constr_1341]	Consistency of <code>DiagnosticServiceSwMapping</code> with respect to asynchronously called <code>DiagnosticRoutines</code>
[constr_1342]	Possible values for <code>DiagnosticSecurityAccess.requestSeedId</code>
[constr_1343]	Simultaneous existence of the attributes <code>DiagnosticServiceDataMapping.diagnosticDataElement</code> and <code>DiagnosticDataByIdentifier.dataIdentifier</code>
[constr_1344]	Condition for the identification of data types of attributes <code>DiagnosticServiceDataMapping.mappedDataElement</code> and <code>DiagnosticServiceDataMapping.diagnosticDataElement</code>
[constr_1345]	<code>DiagnosticDataElement</code> shall not (finally) be aggregated by a <code>DiagnosticRoutine</code>
[constr_1346]	Allowed values of <code>DiagnosticServiceSwMapping.serviceInstance</code>
[constr_1347]	Existence of attributes of <code>DiagnosticServiceSwMapping</code>
[constr_1349]	Value of <code>udsDtcValue</code> shall be unique
[constr_1350]	Value of <code>DiagnosticTroubleCodeGroup.groupNumber</code> shall be unique
[constr_1351]	Value of <code>DiagnosticTroubleCodeGroup.groupNumber</code>
[constr_1352]	Existence of <code>maxNumberFreezeFrameRecords</code> vs. <code>freezeFrame</code>
[constr_1353]	Applicability of [constr_1352]
[constr_1354]	Existence of attribute <code>DiagnosticTroubleCodeProps.freezeFrameContent</code>
[constr_1355]	Value of <code>recordNumber</code>
[constr_1356]	Value of <code>recordNumber</code> shall be unique
[constr_1357]	Value of <code>recordNumber</code>
[constr_1358]	Value of <code>recordNumber</code> shall be unique
[constr_1359]	Existence of attribute <code>DiagnosticDebounceAlgorithmProps.debounceCounterStorage</code>
[constr_1360]	Usage of <code>DiagEventDebounceMonitorInternal</code> is not supported in the context of <code>DiagnosticDebounceAlgorithmProps</code>
[constr_1361]	Number of <code>DiagnosticEventToEnableConditionGroupMapping</code> elements per <code>DiagnosticEvent</code>
[constr_1362]	Number of <code>DiagnosticEventToStorageConditionGroupMapping</code> elements per <code>DiagnosticEvent</code>





Number	Heading
[constr_1365]	Multiplicity of <a href="#">DiagnosticResponseOnEvent.event</a>
[constr_1366]	Event ID in the context of diagnostic service <a href="#">ResponseOnEvent</a> shall be unique
[constr_1376]	Multiplicity of reference <a href="#">DiagnosticTroubleCodeProps.memoryDestination</a>
[constr_1377]	Existence of reference <a href="#">DiagnosticTroubleCodeProps.memoryDestination</a>
[constr_1378]	Value of <a href="#">DiagnosticMemoryDestinationUserDefined.memoryId</a>
[constr_1379]	Existence of <a href="#">DiagnosticMemoryDestinationPrimary</a>
[constr_1380]	Existence of <a href="#">DiagnosticMemoryDestinationMirror</a>

**Table B.2: Added Constraints in R4.2.1**

## B.2 Constraint History of this Document according to AUTOSAR R4.2.2

### B.2.1 Added Traceables in R4.2.2

none

### B.2.2 Changed Traceables in R4.2.2

none

### B.2.3 Deleted Traceables in R4.2.2

none

### B.2.4 Added Constraints in R4.2.2

Number	Heading
[constr_1394]	Value of <a href="#">DiagnosticDataElement.maxNumberOfElements</a> depending on its existence

**Table B.3: Added Constraints in R4.2.2**

### B.2.5 Changed Constraints in R4.2.2

none

## B.2.6 Deleted Constraints in R4.2.2

none

## B.3 Constraint History of this Document according to AUTOSAR R4.3.0

### B.3.1 Added Traceables in R4.3.0

Number	Heading
[TPS_DEXT_01091]	Relation between a <a href="#">DiagnosticServiceTable</a> and one or more <a href="#">DiagnosticConnections</a>
[TPS_DEXT_01092]	Semantics of <a href="#">DiagnosticParameterIdentifier</a>
[TPS_DEXT_01093]	Definition of legislative freeze frame for WWHH-OBD
[TPS_DEXT_01094]	Semantics of meta-class <a href="#">DiagnosticTroubleCodeUdsToTroubleCodeObdMapping</a>
[TPS_DEXT_01095]	Definition of “alias” diagnostic event for the creation of a Fim configuration in the diagnostic extract
[TPS_DEXT_01096]	Semantics of <a href="#">DiagnosticFunctionInhibitSource</a>
[TPS_DEXT_01097]	Standardized value of <a href="#">StructuredReq.category</a> for the modeling of <a href="#">DiagnosticFunctionInhibitSource</a>
[TPS_DEXT_01098]	Semantics of attribute <a href="#">DiagnosticFunctionInhibitSource.event</a>
[TPS_DEXT_01099]	Semantics of attribute <a href="#">DiagnosticFunctionInhibitSource.eventGroup</a>
[TPS_DEXT_01100]	Consequence of the existence of <a href="#">DiagnosticFimAliasEventMapping</a>
[TPS_DEXT_01101]	Consequence of the existence of <a href="#">DiagnosticFimAliasEventGroupMapping</a>
[TPS_DEXT_01102]	Semantics of <a href="#">DiagnosticFimFunctionMapping</a>
[TPS_DEXT_01103]	Semantics of meta-class <a href="#">DiagnosticJ1939SpnMapping</a>
[TPS_DEXT_01104]	Difference between <a href="#">DiagnosticJ1939FreezeFrame</a> and <a href="#">DiagnosticJ1939ExpandedFreezeFrame</a>
[TPS_DEXT_01105]	Relation of <a href="#">DiagnosticJ1939Spn</a> to <a href="#">DiagnosticJ1939FreezeFrame</a> and <a href="#">DiagnosticJ1939ExpandedFreezeFrame</a>
[TPS_DEXT_01106]	Relation of Controller Application to SPN
[TPS_DEXT_01107]	Definition of service-only DTC
[TPS_DEXT_01108]	Purpose of the <a href="#">DiagnosticJ1939SwMapping</a>
[TPS_DEXT_01110]	Standardized values of <a href="#">DiagnosticJumprGroup.category</a>
[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 <a href="#">DiagnosticEnvConditionFormula</a>





Number	Heading
[TPS_DEXT_01114]	<a href="#">DiagnosticEnvConditionFormula</a> that has no <a href="#">parts</a>
[TPS_DEXT_01115]	<a href="#">DiagnosticEnvConditionFormula</a> that has one <a href="#">part</a>
[TPS_DEXT_01116]	<a href="#">DiagnosticEnvConditionFormula</a> that has more than one <a href="#">part</a>
[TPS_DEXT_01117]	Semantics of <a href="#">DiagnosticEnvConditionFormula.nrcValue</a>
[TPS_DEXT_01118]	Semantics of <a href="#">DiagnosticEnvDataCondition</a>
[TPS_DEXT_01119]	Semantics of <a href="#">DiagnosticEnvModeCondition</a>
[TPS_DEXT_01120]	Comparison of the value of a <a href="#">ModeDeclarationGroupPrototype</a> with a <a href="#">ModeDeclaration</a>
[TPS_DEXT_01121]	Semantics of <a href="#">DiagnosticFunctionIdentifier</a>
[TPS_DEXT_01122]	Indication whether a <a href="#">EcuInstance</a> supports OBD
[TPS_DEXT_01124]	Semantics of meta-class <a href="#">DiagnosticProtocol</a>
[TPS_DEXT_01125]	Support for diagnostic service <a href="#">RequestCurrentPowertrainDiagnosticData</a>
[TPS_DEXT_01126]	Support of OBD service <a href="#">RequestPowertrainFreezeFrameData</a>
[TPS_DEXT_01127]	Semantics of meta-class <a href="#">DiagnosticRequestEmissionRelatedDTC</a>
[TPS_DEXT_01128]	Semantics of meta-class <a href="#">DiagnosticClearResetEmissionRelatedInfo</a>
[TPS_DEXT_01129]	Support for OBD diagnostic service <a href="#">RequestOnBoardMonitoringTestResults</a>
[TPS_DEXT_01130]	Support of OBD diagnostic service <a href="#">RequestControlOfOnBoardDevice</a>
[TPS_DEXT_01131]	Support for OBD diagnostic service <a href="#">RequestVehicleInformation</a>
[TPS_DEXT_01132]	Support for OBD diagnostic service <a href="#">RequestEmissionRelatedDiagnosticTroubleCodesPermanentStatus</a>
[TPS_DEXT_01133]	Support for WWH-OBD within the diagnostic extract
[TPS_DEXT_01134]	Definition of a <a href="#">DiagnosticDataElement</a> used in the context of a DID obtained by sender-receiver communication
[TPS_DEXT_01135]	Definition of a <a href="#">DiagnosticDataElement</a> used in the context of a DID obtained by client/server communication
[TPS_DEXT_01136]	Definition of a <a href="#">DiagnosticDataElement</a> used in the context of a diagnostic routine
[TPS_DEXT_01137]	Applicability of <a href="#">DiagnosticDataIdentifier.didSize</a>
[TPS_DEXT_01138]	Applicability of <a href="#">DiagnosticDataIdentifier.supportInfoByte</a>
[TPS_DEXT_01139]	Semantics of the references from <a href="#">DiagnosticAccessPermission</a>

**Table B.4: Added Traceables in 4.3.0**



### B.3.2 Changed Traceables in R4.3.0

Number	Heading
[TPS_DEXT_01006]	The role of <code>DiagnosticServiceTables</code>
[TPS_DEXT_01052]	Existence of attribute <code>DiagnosticServiceInstance.accessPermission</code>
[TPS_DEXT_01060]	Applicable values for <code>DiagnosticReadDTCInformation.category</code>
[TPS_DEXT_01074]	Difference between the attributes <code>DiagnosticComControlClass.specificChannel</code> and <code>DiagnosticComControlClass.subNodeChannel</code>
[TPS_DEXT_03003]	Semantics of <code>DiagnosticEventToTroubleCodeUdsMapping</code>

**Table B.5: Changed Traceables in R4.3.0**

### B.3.3 Deleted Traceables in R4.3.0

Number	Heading
[TPS_DEXT_01058]	Standardized values for <code>DiagnosticDynamicallyDefineDataIdentifier.category</code>
[TPS_DEXT_01059]	Applicable values for <code>DiagnosticPeriodicRate.category</code>

**Table B.6: Deleted Traceables in R4.3.0**

### B.3.4 Added Constraints in R4.3.0

Number	Heading
[constr_1405]	Value of <code>DiagnosticProtocol.serviceTable</code> vs. <code>DiagnosticServiceTable.protocolKind</code>
[constr_1406]	<code>DiagnosticServiceTable.diagnosticConnection</code> vs. <code>DiagnosticProtocol.diagnosticConnection</code>
[constr_1411]	Existence of <code>DiagnosticMemoryIdentifier.memoryHighAddressLabel</code> vs. <code>DiagnosticMemoryIdentifier.memoryHighAddress</code>
[constr_1412]	Existence of <code>DiagnosticMemoryIdentifier.memoryLowAddressLabel</code> vs. <code>DiagnosticMemoryIdentifier.memoryLowAddress</code>
[constr_1419]	Value of <code>DiagnosticSecurityLevel.accessDataRecordSize</code>
[constr_1421]	Consistency of <code>DiagnosticDynamicallyDefineDataIdentifierClass.subfunction</code>
[constr_1435]	Debouncing in the presence of a <code>DiagnosticEventPortMapping</code>
[constr_1447]	Restrictions for the value of <code>DiagnosticParameterIdentifier.id</code>
[constr_1448]	Interval of <code>DiagnosticParameterIdentifier.id</code>
[constr_1449]	PID shall only carry a fixed-length collection of data
[constr_1450]	Service mapping for OBD mode 0x01 for <code>DiagnosticParameterIdentifier</code>
[constr_1451]	Service mapping for OBD mode 0x09 for <code>DiagnosticInfoType</code>
[constr_1452]	Service mapping for OBD mode 0x08 for <code>DiagnosticInfoType</code>







Number	Heading
[constr_1453]	References from <code>DiagnosticFunctionInhibitSource</code>
[constr_1454]	<code>DiagnosticFimFunctionMapping</code> shall only reference a <code>SwcServiceDependency</code> that aggregates <code>FunctionInhibitionNeeds</code>
[constr_1455]	Relation of <code>DiagnosticJ1939Node</code> to <code>J1939NmNode</code>
[constr_1456]	Valid interval for attribute <code>DiagnosticTroubleCodeJ1939.fmi</code>
[constr_1457]	Service-only DTCs shall refer to a common memory section
[constr_1458]	Reference to <code>DiagnosticMemoryDestination</code>
[constr_1459]	Existence of attributes of <code>DiagnosticTroubleCodeProps</code>
[constr_1460]	Restrictions for the value of <code>DiagnosticInfoType.id</code>
[constr_1461]	Restrictions for the value of <code>DiagnosticTestRoutineIdentifier.id</code>
[constr_1462]	Restrictions for the value of <code>DiagnosticTestResult.testIdentifier.id</code>
[constr_1464]	Allowed value range of <code>DiagnosticEnvConditionFormula.nrcValue</code>
[constr_1465]	Allowed values of <code>compareType</code> in the context of a <code>DiagnosticEnvDataCondition</code>
[constr_1466]	Allowed values of <code>compareType</code> in the context of a <code>DiagnosticEnvModeCondition</code>
[constr_1467]	References in <code>DiagnosticEnvModeCondition</code>
[constr_1470]	Value of <code>DiagnosticParameter.bitOffset</code>
[constr_1471]	Existence of <code>DiagnosticDataIdentifier.didSize</code>
[constr_1472]	Existence of <code>DiagnosticDataIdentifier.supportInfoByte</code>

**Table B.7: Added Constraints in R4.3.0**

### B.3.5 Changed Constraints in R4.3.0

Number	Heading
[constr_1325]	Allowed attributes of <code>SwDataDefProps</code> for <code>DiagnosticDataElement.swDataDefProps</code>
[constr_1327]	Multiplicity of <code>DiagnosticEcuInstanceProps.ecuInstance</code>
[constr_1328]	Consistency of <code>DiagnosticEcuInstanceProps.ecuInstance</code> and <code>DiagnosticServiceTable.ecuInstance</code>

**Table B.8: Changed Constraints in R4.3.0**

### B.3.6 Deleted Constraints in R4.3.0

none

## B.4 Constraint History of this Document according to AUTOSAR R4.3.1

### B.4.1 Added Traceables in 4.3.1

Number	Heading
[TPS_DEXT_01140]	Values contained in <a href="#">DiagnosticExtract</a> shall be taken for the derivation of basic software modules

**Table B.9: Added Traceables in 4.3.1**

### B.4.2 Changed Traceables in 4.3.1

Number	Heading
[TPS_DEXT_01087]	Semantics of <a href="#">DiagnosticOperationCycle</a>

**Table B.10: Changed Traceables in 4.3.1**

### B.4.3 Deleted Traceables in 4.3.1

Number	Heading
[TPS_DEXT_01073]	Diagnostic properties that are specific to an individual <a href="#">EcuInstance</a>

**Table B.11: Deleted Traceables in 4.3.1**

### B.4.4 Added Constraints in 4.3.1

Number	Heading
[constr_1509]	<a href="#">extendedDataRecord.recordNumber</a> shall be unique within primary fault memory
[constr_1510]	<a href="#">extendedDataRecord.recordNumber</a> shall be unique within mirror fault memory
[constr_1511]	<a href="#">extendedDataRecord.recordNumber</a> shall be unique within user-defined fault memory
[constr_1512]	<a href="#">freezeFrame.recordNumber</a> shall be unique within primary fault memory
[constr_1513]	<a href="#">freezeFrame.recordNumber</a> shall be unique within mirror fault memory
[constr_1514]	<a href="#">freezeFrame.recordNumber</a> shall be unique within user-defined fault memory
[constr_1515]	Reference from <a href="#">DiagnosticRoutineControl</a> to <a href="#">DiagnosticAccessPermission</a> has no meaning

**Table B.12: Added Constraints in 4.3.1**

### B.4.5 Changed Constraints in 4.3.1

Number	Heading
[constr_1325]	Allowed attributes of <code>SwDataDefProps</code> for <code>DiagnosticDataElement.sw-DataDefProps</code>
[constr_1450]	Service mapping for ODB mode 0x01 for <code>DiagnosticParameterIdentifier</code>
[constr_1451]	Service mapping for ODB mode 0x09 for <code>DiagnosticInfoType</code>

**Table B.13: Changed Constraints in 4.3.1**

### B.4.6 Deleted Constraints in 4.3.1

Number	Heading
[constr_1356]	Value of <code>recordNumber</code> shall be unique
[constr_1358]	Value of <code>recordNumber</code> shall be unique

**Table B.14: Deleted Constraints in 4.3.1**

## B.5 Constraint History of this Document according to AUTOSAR R4.4.0

### B.5.1 Added Traceables in 4.4.0

Number	Heading
[TPS_DEXT_01141]	Definition of a collection of test results
[TPS_DEXT_01142]	Standardized values for <code>DiagnosticDemProvidedDataMapping.dataProvider</code>
[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 <code>DiagnosticEventToTrouble-CodeJ1939Mapping</code>
[TPS_DEXT_01146]	Support for service <code>ReadScalingDataByIdentifier</code> (0x24)

**Table B.15: Added Traceables in 4.4.0**

### B.5.2 Changed Traceables in 4.4.0

Number	Heading
[TPS_DEXT_01004]	<code>DiagnosticContributionSet</code> defines the scope of the <code>DiagnosticExtract</code>
[TPS_DEXT_01041]	Semantics of attribute <code>DiagnosticServiceDataMapping.diagnosticDataElement</code>
[TPS_DEXT_01055]	Standardized values of <code>DiagnosticContributionSet.category</code>
[TPS_DEXT_01079]	Modeling of the arguments to a <code>DiagnosticRoutine</code>
[TPS_DEXT_03011]	Clearing request for a <code>DiagnosticEvent</code>
[TPS_DEXT_03020]	Semantics of <code>DiagnosticDemProvidedDataMapping</code>

**Table B.16: Changed Traceables in 4.4.0**

### B.5.3 Deleted Traceables in 4.4.0

Number	Heading
[TPS_DEXT_01137]	Applicability of <code>DiagnosticDataIdentifier.didSize</code>

**Table B.17: Deleted Traceables in 4.4.0**

### B.5.4 Added Constraints in 4.4.0

Number	Heading
[constr_1552]	<code>DiagnosticDataIdentifier</code> referenced by <code>DiagnosticDataIdentifierSet</code>
[constr_1584]	<code>DiagnosticDataElement</code> shall not be used more than once in I/O Control instance
[constr_1590]	<code>DiagnosticEvent</code> referenced in the role <code>masterEvent</code> or <code>slaveEvent</code>
[constr_1591]	<code>DiagnosticEvent</code> referenced as <code>slaveEvent</code> shall not be reported by diagnostic monitor
[constr_1612]	Reference from <code>DiagnosticRoutineControl</code> to <code>DiagnosticAccessPermission</code> has no meaning
[constr_1616]	Existence of attribute <code>DiagnosticExtendedDataRecord.customTrigger</code>
[constr_1617]	Existence of attribute <code>DiagnosticFreezeFrame.customTrigger</code>
[constr_1623]	Restriction on <code>DiagnosticReadScalingDataByIdentifier.dataIdentifier</code>
[constr_1624]	Existence of <code>DiagnosticDataElement.scalingInfoSize</code>
[constr_1633]	Existence of <code>DiagnosticResponseOnEvent.event</code> vs. <code>DiagnosticResponseOnEvent.responseOnEventAction</code>

**Table B.18: Added Constraints in 4.4.0**

### B.5.5 Changed Constraints in 4.4.0

Number	Heading
[constr_1325]	Allowed attributes of <code>SwDataDefProps</code> for <code>DiagnosticDataElement.swDataDefProps</code>
[constr_1328]	Consistency of <code>DiagnosticEcuInstanceProps.ecuInstance</code> and <code>DiagnosticServiceTable.ecuInstance</code>

**Table B.19: Changed Constraints in 4.4.0**

### B.5.6 Deleted Constraints in 4.4.0

Number	Heading
[constr_1471]	Existence of <code>DiagnosticDataIdentifier.didSize</code>
[constr_1515]	Reference from <code>DiagnosticRoutineControl</code> to <code>DiagnosticAccessPermission</code> has no meaning

**Table B.20: Deleted Constraints in 4.4.0**

## C Glossary

**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 ([21]).

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: `EcucParameterDef`, `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 `<<atpSplittable>>`. 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.

**Profile Authoring Support Data** Data that is used for efficient authoring of a profile. E.g. list of referable constraints, meta-classes, meta-attributes or other reusable model assets (blueprints)

**Profile Authoring Tool** A specialized AUTOSAR Tool which focuses on the authoring of profiles for data exchange points. It e.g. provides support for the creation of profiles from scratch, modification of existing profiles or composition of existing profiles.

**Profile Compatibility Checker Tool** A specialized AUTOSAR Tool which focuses on checking the compatibility of profiles for data exchange. Note that this compatibility check includes manual compatibility checks by engineers and automated assistance using more formal algorithms.

**Profile Consistency Checker Tool** A specialized AUTOSAR Tool which focuses on checking the consistency of profiles.

**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`.

## D Modeling of InstanceRef

### D.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 [22].

## D.2 Modeling

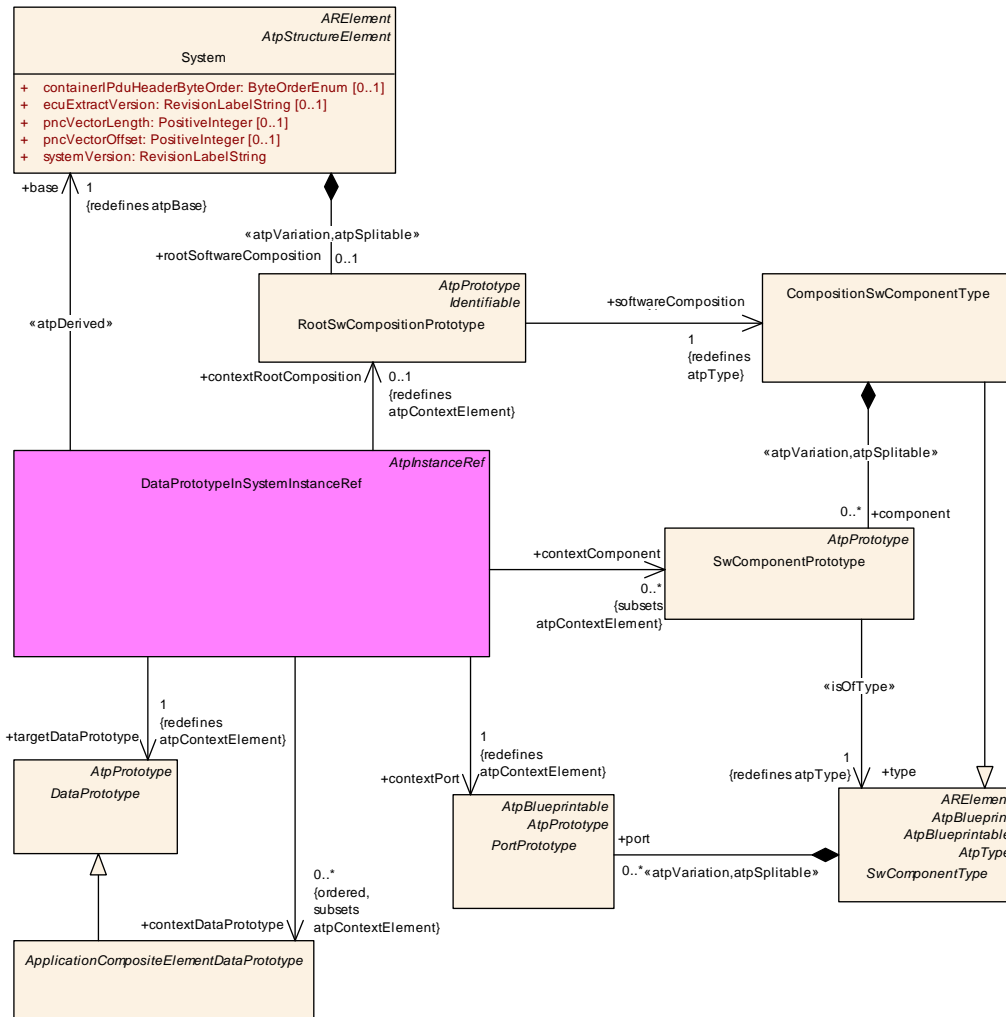
<b>Class</b>	<b>DataPrototypeInSystemInstanceRef</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::InstanceRefs			
<b>Note</b>				
<b>Base</b>	<i>ARObject</i> , <i>AtpInstanceRef</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
base	<a href="#">System</a>	1	ref	This represents the base of the InstanceRef <b>Stereotypes:</b> atpDerived <b>Tags:</b> xml.sequenceOffset=10
context Component	<a href="#">SwComponent Prototype</a>	*	ref	<b>Tags:</b> xml.sequenceOffset=30



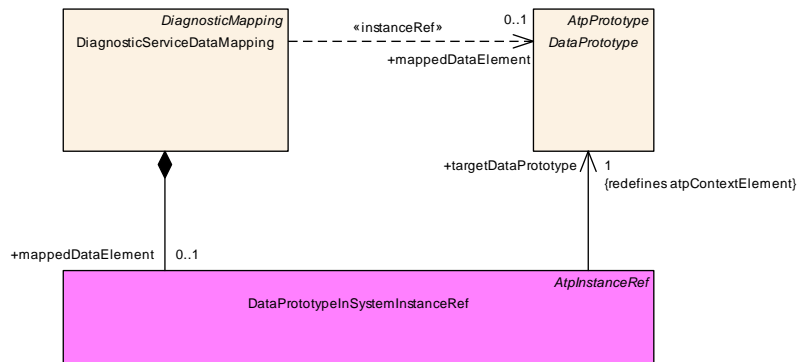


Class	DataPrototypeInSystemInstanceRef			
contextDataPrototype (ordered)	ApplicationCompositeElementDataPrototype	*	ref	<b>Tags:</b> xml.sequenceOffset=50
contextPort	PortPrototype	1	ref	This represents the PortPrototype that is contained in the InstanceRef. <b>Tags:</b> xml.sequenceOffset=40
contextRootComposition	RootSwCompositionPrototype	0..1	ref	<b>Tags:</b> xml.sequenceOffset=20
targetDataPrototype	DataPrototype	1	ref	This represents the target of the InstanceRef <b>Tags:</b> xml.sequenceOffset=60

**Table D.1: DataPrototypeInSystemInstanceRef**



**Figure D.1: Modeling of DataPrototypeInSystemInstanceRef**



**Figure D.2: Modeling of DiagnosticServiceDataMapping**

<b>Class</b>	<b>SwcServiceDependencyInSystemInstanceRef</b>			
<b>Package</b>	M2::AUTOSARTemplates::DiagnosticExtract::InstanceRefs			
<b>Note</b>				
<b>Base</b>	ARObject, AtpInstanceRef			
<b>Attribute</b>	<b>Type</b>	<b>Mul.</b>	<b>Kind</b>	<b>Note</b>
contextRootSw Composition	RootSwComposition Prototype	1	ref	
contextSw Component Prototype	SwComponent Prototype	*	ref	
targetSwc Service Dependency	SwcService Dependency	1	ref	

**Table D.2: SwcServiceDependencyInSystemInstanceRef**

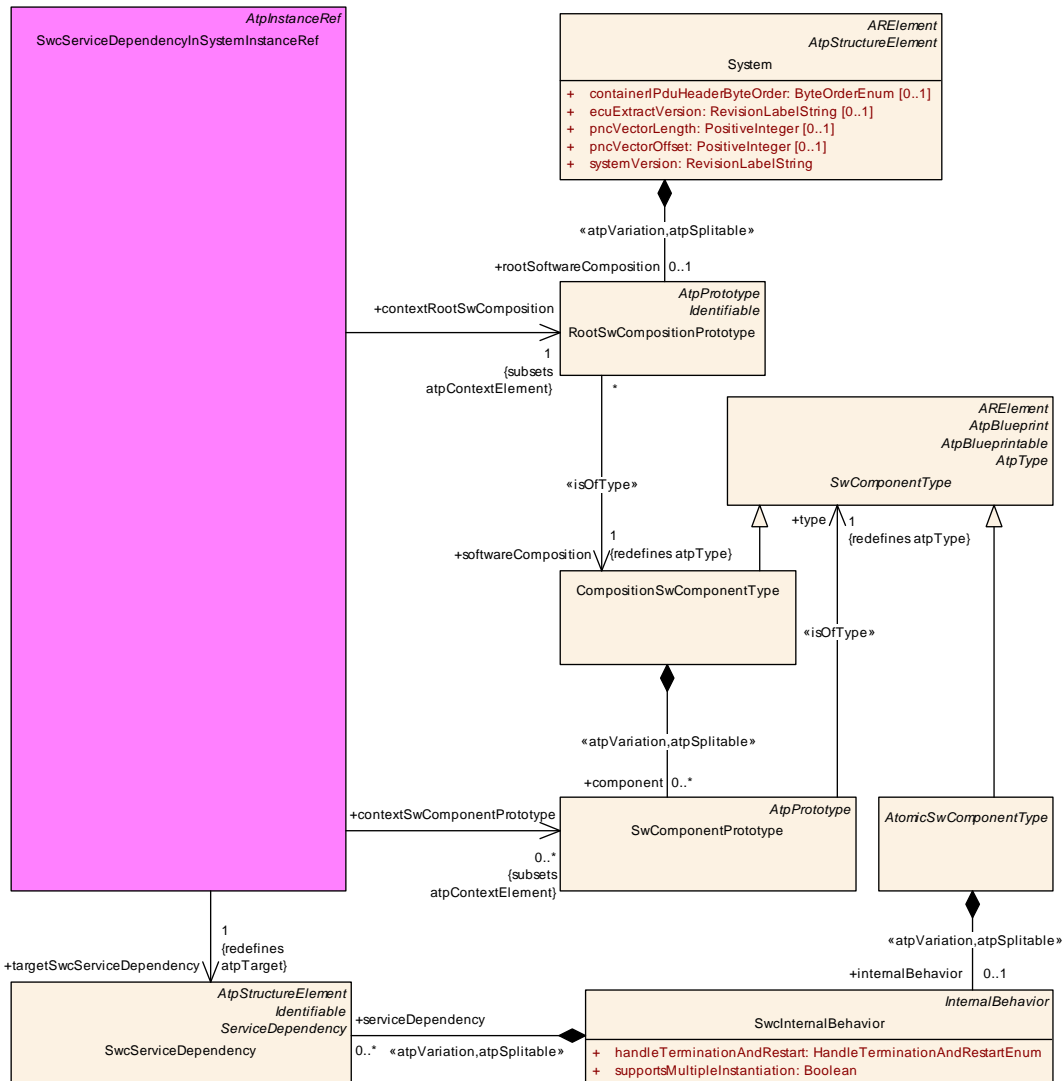


Figure D.3: Modeling of `SwcServiceDependencyInSystemInstanceRef`

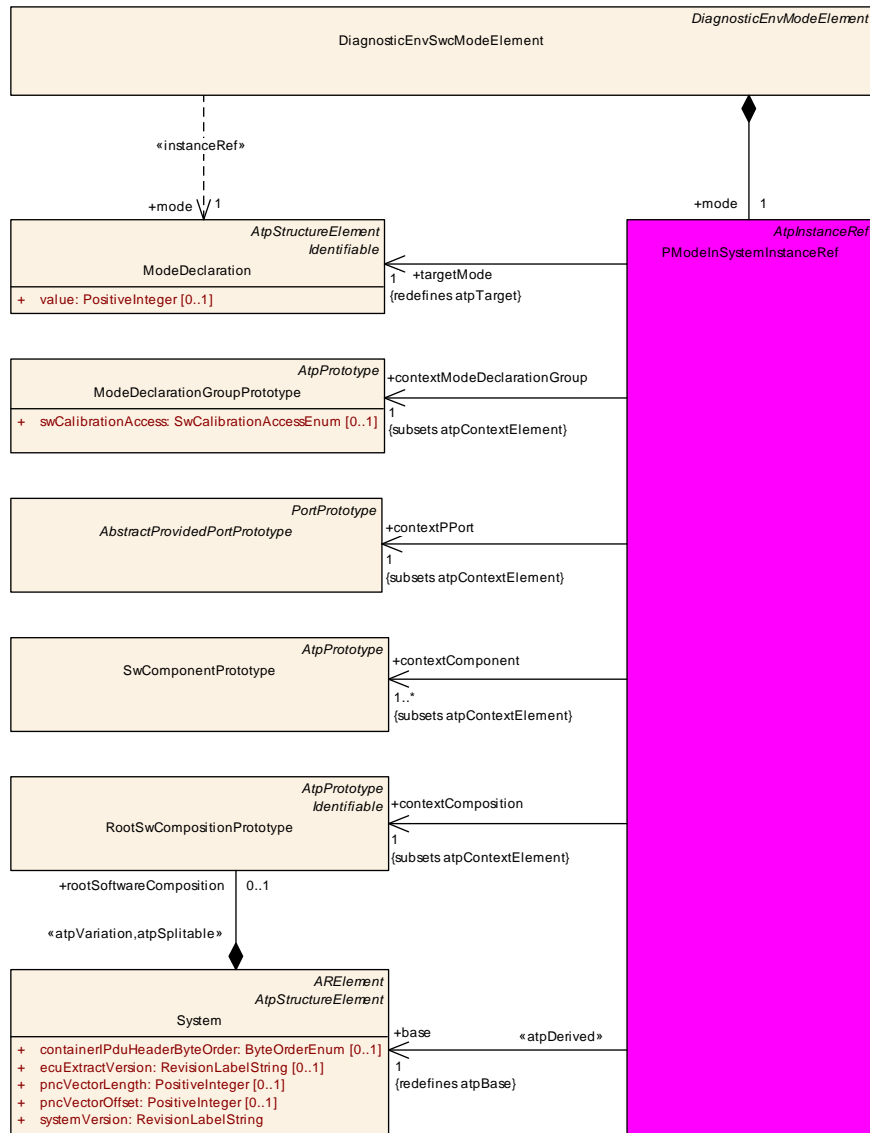
Class	PModelInSystemInstanceRef			
Package	M2::AUTOSARTemplates::DiagnosticExtract::InstanceRefs			
Note				
Base	ARObject, AtpInstanceRef			
Attribute	Type	Mul.	Kind	Note
base	System	1	ref	<b>Stereotypes:</b> atpDerived <b>Tags:</b> xml.sequenceOffset=10
context Component	SwComponent Prototype	1..*	ref	<b>Tags:</b> xml.sequenceOffset=30
context Composition	RootSwComposition Prototype	1	ref	<b>Tags:</b> xml.sequenceOffset=20
contextMode Declaration Group	ModeDeclarationGroup Prototype	1	ref	<b>Tags:</b> xml.sequenceOffset=50





Class	PModelInSystemInstanceRef			
contextPPort	AbstractProvidedPort Prototype	1	ref	Tags: xml.sequenceOffset=40
targetMode	ModeDeclaration	1	ref	Tags: xml.sequenceOffset=60

**Table D.3: PModelInSystemInstanceRef**



**Figure D.4: Formal modeling of the comparison of a ModeDeclarationGroupPrototype with a ModeDeclaration**

## E Upstream Mapping

### E.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 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 tables contain the following columns:

**bsw module:** Name of BSW module

**bsw context:** Reference to parameter container

**bsw type:** Type of parameter

**bsw param:** Name of the BSW parameter

**bsw desc:** Description from the configuration document

**m2 template:** System Template, SW Component Template, ECU Resource Template

**m2 param:** Name of the upstream template parameter

**m2 description:** Description from the upstream template definition

**mapping rule:** Textual description on how to transform between M2 and BSW domains

**mapping type:**

- 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

### E.2 Dcm

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsd	
BSW Parameter		BSW Type
DcmDsdServiceTable		EcucParamConfContainerDef

<b>BSW Description</b>	
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.	
<b>Template Description</b>	
This meta-class represents a model of a diagnostic service table, i.e. the UDS services applicable for a given ECU.	
<b>M2 Parameter</b>	
DiagnosticExtract::DiagnosticContribution::DiagnosticServiceTable	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00040

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDsdSidTabSecurityLevelRef	EcucReferenceDef
<b>BSW Description</b>	
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.	
<b>Template Description</b>	
This represents the associated DiagnosticSecurityLevels	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticAccessPermission.securityLevel	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00041

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDsdSidTabServiceId	EcucIntegerParamDef
<b>BSW Description</b>	
Identifier of the service.	
The possible service identifiers are defined in ISO 14229-1 and ISO 15031-5.	
<b>Template Description</b>	
This meta-class provides the ability to define common properties that are shared among all instances of sub-classes of DiagnosticServiceInstance.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::CommonService::DiagnosticServiceClass	
<b>Mapping Rule</b>	<b>Mapping Type</b>
Service identifiers of the used DiagnosticServiceClass	full
<b>Mapping Status</b>	<b>Mapping ID</b>

valid	up_Dcm_00042
-------	--------------

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService	
BSW Parameter	BSW Type	
DcmDsdSidTabSessionLevelRef	EcucReferenceDef	
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		
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00043	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService	
BSW Parameter	BSW Type	
DcmDsdSidTabSubfuncAvail	EcucBooleanParamDef	
BSW Description		
Information about whether the service has subfunctions or not. This parameter is used for the handling of the "suppressPosRspMsgIndicationBit" as defined in ISO 14229-1, which can be used as a reference for the configuration.		
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		
GenericStructure::GeneralTemplateClasses::Identifiable::Identifiable.category		
Mapping Rule	Mapping Type	
Standardized 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	Mapping ID	
valid	up_Dcm_00044	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService/DcmDsdSubService	
BSW Parameter	BSW Type	



DcmDsdSubServiceFnc	EcucFunctionNameDef
<b>BSW Description</b>	
<p>Callback function of the ECU Supplier specific component for the particular service. The function's prototype is as described for &lt;Module&gt;_&lt;DiagnosticService&gt;_&lt;SubService&gt;.</p> <p>If this parameter is not configured, the subservice is handled Dcm-internally.</p>	
<b>Template Description</b>	
<p>This attribute shall be used to define a custom sub-function number if none of the standardized values of category shall be used.</p>	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl::DiagnosticComControl.customSubFunctionNumber	
<b>Mapping Rule</b>	<b>Mapping Type</b>
The existence of a custom subfunction number shall trigger the creation of a custom processor.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00286

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService/DcmDsdSubService	
<b>BSW Parameter</b>	<b>BSW Type</b>	
DcmDsdSubServiceId	EcucIntegerParamDef	
<b>BSW Description</b>		
<p>Identifier of the subservice.</p> <p>The possible subservice identifiers are defined in ISO 14229-1 and ISO 15031-5.</p>		
<b>Template Description</b>		
<p>The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.</p>		
<b>M2 Parameter</b>		
GenericStructure::GeneralTemplateClasses::Identifiable::Identifiable.category		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
Numerical values of diagnostic service according to ISO 14229 correspond to values of DiagnosticServiceInstance.category.	partial	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid	up_Dcm_00045	

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService/DcmDsdSubService	
<b>BSW Parameter</b>	<b>BSW Type</b>	
DcmDsdSubServiceSecurityLevelRef	EcucReferenceDef	
<b>BSW Description</b>		
<p>Reference to a Security Level in which the subservice is allowed to be executed. Multiple references are allowed for a subservice.</p> <p>Please refer to ISO 14229-1, ISO 15031-5 and chapter "Verification of the Service Security Access levels."</p> <p>If there is no reference configured, no subservice security verification shall be performed.</p>		
<b>Template Description</b>		

This represents the associated DiagnosticSecurityLevels	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticAccessPermission.securityLevel	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00046

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService/DcmDsdSubService
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDsdSubServiceSessionLevelRef	EcucReferenceDef
<b>BSW Description</b>	
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.	
<b>Template Description</b>	
This represents the associated DiagnosticSessions	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00047

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslDiagResp
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDslDiagRespMaxNumRespPend	EcucIntegerParamDef
<b>BSW Description</b>	
Maximum number of negative responses with response code 0x78 (requestCorrectlyReceivedResponsePending) allowed for a request. If Dcm reaches this limit, an automatic 0x10 (generalReject) final response will be transmitted and the service processing will be cancelled.	
<b>Template Description</b>	
Maximum number of negative responses with response code 0x78 (requestCorrectlyReceivedResponsePending) 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.	
<b>M2 Parameter</b>	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.maxNumberOfRequestCorrectlyReceivedResponsePending	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00048

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslDiagResp	
<b>BSW Parameter</b>	<b>BSW Type</b>	
DcmDslDiagRespOnSecondDeclinedRequest	EcucBooleanParamDef	
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
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.		
<b>M2 Parameter</b>		
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.responseOnSecondDeclinedRequest		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00049

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol	
<b>BSW Parameter</b>	<b>BSW Type</b>	
DcmDslProtocolRow	EcucParamConfContainerDef	
<b>BSW Description</b>		
This container contains the configuration of one particular diagnostic protocol used in Dcm.		
<b>Template Description</b>		
This meta-class represents the ability to define a diagnostic protocol.		
<b>M2 Parameter</b>		
DiagnosticExtract::DiagnosticContribution::DiagnosticProtocol		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00050

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection	
<b>BSW Parameter</b>	<b>BSW Type</b>	
DcmDslProtocolComMChannelRef	EcucReferenceDef	
<b>BSW Description</b>		
Reference to the ComMChannel on which the DcmDslProtocolRxPdu is received and the DcmDslProtocolTxPdu is transmitted.		
<b>Template Description</b>		

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.

<b>M2 Parameter</b>	
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationCluster	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00051

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection/DcmDslProtocolRx/DcmDslProtocolRxAddrType
<b>BSW Parameter</b>	<b>BSW Type</b>
DCM_FUNCTIONAL_TYPE	EcucEnumerationLiteralDef
<b>BSW Description</b>	
FUNCTIONAL = 1 to n communication	
<b>Template Description</b>	
Reference to functional request messages.	
<b>M2 Parameter</b>	
SystemTemplate::DiagnosticConnection::DiagnosticConnection.functionalRequest	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00052

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection/DcmDslProtocolRx/DcmDslProtocolRxAddrType
<b>BSW Parameter</b>	<b>BSW Type</b>
DCM_PHYSICAL_TYPE	EcucEnumerationLiteralDef
<b>BSW Description</b>	
PHYSICAL = 1 to 1 communications using physical addressing	
<b>Template Description</b>	
Reference to a physical request message.	
<b>M2 Parameter</b>	
SystemTemplate::DiagnosticConnection::DiagnosticConnection.physicalRequest	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00053

<b>BSW Module</b>	<b>BSW Context</b>
-------------------	--------------------

Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection/DcmDslProtocolRx	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDslProtocolRxPduRef	EcucReferenceDef	
<b>BSW Description</b>		
Reference to a Pdu in EcuC that is used for this reception channel.		
<b>Template Description</b>		
The IPdu (Interaction Layer Protocol Data Unit) element is used to sum up all Pdus that are routed by the PduR.		
<b>M2 Parameter</b>		
SystemTemplate::Fibex::FibexCore::CoreCommunication::IPdu		
<b>Mapping Rule</b>		<b>Mapping Type</b>
Reference to IPdu of xxxTpConnection for DiagnosticConnection.physicalRequest / DiagnosticConnection.functionalRequest		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00054

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDslProtocolRxConnectionId	EcucIntegerParamDef	
<b>BSW Description</b>		
Unique identifier of the tester which uses this connection for diagnostic communication.		
<b>Template Description</b>		
An ECU specific ID for responses of diagnostic routines.		
<b>M2 Parameter</b>		
SystemTemplate::Fibex::FibexCore::CoreTopology::EcuInstance.diagnosticAddress		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00055

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDslProtocolRxTesterSourceAddr	EcucIntegerParamDef	
<b>BSW Description</b>		
Tester source address uniquely describes a client and will be used e.g within the jump to Bootloader interfaces. This parameter is not required for generic connections (DcmPdus with MetaDataLength >= 1).		
<b>Template Description</b>		
An ECU specific ID for responses of diagnostic routines.		
<b>M2 Parameter</b>		
SystemTemplate::Fibex::FibexCore::CoreTopology::EcuInstance.diagnosticAddress		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection	
BSW Parameter		BSW Type
DcmDslProtocolTx		EcucParamConfContainerDef
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		
SystemTemplate::DiagnosticConnection::DiagnosticConnection.response		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00056

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection/DcmDslProtocolTx	
BSW Parameter		BSW Type
DcmDslProtocolTxPduRef		EcucReferenceDef
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		
SystemTemplate::Fibex::FibexCore::CoreCommunication::IPdu		
Mapping Rule		Mapping Type
Reference to IPdu of xxxTpConnection for DiagnosticConnection.response		full
Mapping Status		Mapping ID
valid		up_Dcm_00057

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslPeriodicTransmission/DcmDslPeriodicConnection	
BSW Parameter		BSW Type
DcmDslPeriodicTxPduRef		EcucReferenceDef
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		
SystemTemplate::Fibex::FibexCore::CoreCommunication::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
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00058

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslResponseOnEvent	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDslRoeTxPduRef		EcucReferenceDef
<b>BSW Description</b>		
Reference to a Pdu in EcuC that is used for this ResponseOnEvent transmission connection.		
<b>Template Description</b>		
The IPdu (Interaction Layer Protocol Data Unit) element is used to sum up all Pdus that are routed by the PduR.		
<b>M2 Parameter</b>		
SystemTemplate::Fibex::FibexCore::CoreCommunication::IPdu		
<b>Mapping Rule</b>		<b>Mapping Type</b>
Reference to IPdu of xxxTpConnection for DiagnosticConnection.responseOnEvent		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00059

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDslProtocolPriority		EcucIntegerParamDef
<b>BSW Description</b>		
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		
<b>Template Description</b>		
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		
<b>M2 Parameter</b>		
DiagnosticExtract::DiagnosticContribution::DiagnosticProtocol.priority		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00068

<b>BSW Module</b>	<b>BSW Context</b>
-------------------	--------------------

Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolTransType	
<b>BSW Parameter</b>		<b>BSW Type</b>
TYPE1		EcucEnumerationLiteralDef
<b>BSW Description</b>		
Messages on the DcmTxPduld already used for normal diagnostic responses. The outgoing messages must be synchronized with 'normal outgoing messages', which have a higher priority.		
<b>Template Description</b>		
Reference to a ROE message.		
<b>M2 Parameter</b>		
SystemTemplate::DiagnosticConnection::DiagnosticConnection.responseOnEvent		
<b>Mapping Rule</b>		<b>Mapping Type</b>
TYPE1 : periodicResponseTp / responseOnEvent using same reference as the normal response		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00069

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolTransType	
<b>BSW Parameter</b>		<b>BSW Type</b>
TYPE2		EcucEnumerationLiteralDef
<b>BSW Description</b>		
Messages on a separate DcmTxPduld.		
<b>Template Description</b>		
Reference to a ROE message.		
<b>M2 Parameter</b>		
SystemTemplate::DiagnosticConnection::DiagnosticConnection.responseOnEvent		
<b>Mapping Rule</b>		<b>Mapping Type</b>
TYPE2: periodicResponseTp / responseOnEvent using other reference as the normal response		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00070

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolType	
<b>BSW Parameter</b>		<b>BSW Type</b>
DCM_PERIODICTRANS_ON_CAN		EcucEnumerationLiteralDef
<b>BSW Description</b>		
<b>Template Description</b>		
<b>AbstractCanPhysicalChannel:</b> Abstract class that is used to collect the common TtCAN and CAN PhysicalChannel attributes.		
<b>DiagnosticConnection.periodicResponseUudt:</b> Reference to UUDT responses.		
<b>M2 Parameter</b>		
SystemTemplate::Fibex::Fibex4Can::CanTopology::AbstractCanPhysicalChannel, SystemTemplate::DiagnosticConnection::DiagnosticConnection.periodicResponseUudt,		
<b>Mapping Rule</b>		<b>Mapping Type</b>



If DiagnosticConnection.periodicResponseUudt exists and PhysicalChannel given as AbstractCanPhysicalChannel.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00060

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolType	
BSW Parameter	BSW Type	
DCM_PERIODICTRANS_ON_IP	EcucEnumerationLiteralDef	
BSW Description		
Template Description		
<p><b>SocketConnection:</b> The SoAd serves as a (De)Multiplexer between different PDU sources and the TCP/IP stack.</p> <p><b>DiagnosticConnection.periodicResponseUudt:</b> Reference to UUDT responses.</p> <p><b>EthernetPhysicalChannel:</b> The EthernetPhysicalChannel represents a VLAN or an untagged channel. An untagged channel is modeled as an EthernetPhysicalChannel without an aggregated VLAN.</p>		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection, SystemTemplate::DiagnosticConnection::DiagnosticConnection.periodicResponseUudt, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetPhysicalChannel		
Mapping Rule	Mapping Type	
If DiagnosticConnection.periodicResponseUudt exists and PhysicalChannel given as EthernetPhysicalChannel.	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00061	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolType	
BSW Parameter	BSW Type	
DCM_ROE_ON_CAN	EcucEnumerationLiteralDef	
BSW Description		
Template Description		
<p><b>DiagnosticConnection.responseOnEvent:</b> Reference to a ROE message.</p> <p><b>CanTpConnection:</b> A connection identifies the sender and the receiver of this particular communication. The CanTp module routes a Pdu through this connection.</p> <p>atpVariation: Derived, because TpNode can vary.</p>		
M2 Parameter		
SystemTemplate::DiagnosticConnection::DiagnosticConnection.responseOnEvent, SystemTemplate::TransportProtocols::CanTpConnection		
Mapping Rule	Mapping Type	

In case DiagnosticConnection.responseOnEvent exists and TpConnection Ident.ident belongs to a CanTpConnection.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00062

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolType	
BSW Parameter		BSW Type
DCM_ROE_ON_FLEXRAY		EcucEnumerationLiteralDef
BSW Description		
<b>Template Description</b>		
<p><b>DiagnosticConnection.responseOnEvent:</b> Reference to a ROE message.</p> <p><b>FlexrayTpConnection:</b> A connection identifies the sender and the receiver of this particular communication. The FlexRayTp module routes a Pdu through this connection.</p> <p>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.</p>		
M2 Parameter		
SystemTemplate::DiagnosticConnection::DiagnosticConnection.responseOnEvent, SystemTemplate::TransportProtocols::FlexrayTpConnection		
Mapping Rule		Mapping Type
In case DiagnosticConnection.responseOnEvent exists and TpConnection Ident.ident belongs to FlexRayTpConnection		full
Mapping Status		Mapping ID
valid		up_Dcm_00063

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolType	
BSW Parameter		BSW Type
DCM_ROE_ON_IP		EcucEnumerationLiteralDef
BSW Description		
<b>Template Description</b>		
<p><b>DiagnosticConnection.responseOnEvent:</b> Reference to a ROE message.</p> <p><b>SocketConnection:</b> The SoAd serves as a (De)Multiplexer between different PDU sources and the TCP/IP stack.</p>		
M2 Parameter		
SystemTemplate::DiagnosticConnection::DiagnosticConnection.responseOnEvent, SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection		
Mapping Rule		Mapping Type

In case DiagnosticConnection.responseOnEvent exists and TpConnection Ident.ident belongs to SocketConnection	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00064

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolType	
BSW Parameter	BSW Type	
DCM_UDS_ON_CAN	EcucEnumerationLiteralDef	
BSW Description	UDS on CAN (ISO15765-3; ISO14229-1)	
Template Description	<p><b>DiagnosticConnection.physicalRequest:</b> Reference to a physical request message.</p> <p><b>CanTpConnection:</b> A connection identifies the sender and the receiver of this particular communication. The CanTp module routes a Pdu through this connection.</p> <p>atpVariation: Derived, because TpNode can vary.</p>	
M2 Parameter	SystemTemplate::DiagnosticConnection::DiagnosticConnection.physicalRequest, SystemTemplate::TransportProtocols::CanTpConnection	
Mapping Rule	Mapping Type	
In case DiagnosticConnection.physicalRequest exists and TpConnection Ident.ident belongs to CanTpConnection	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00065	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolType	
BSW Parameter	BSW Type	
DCM_UDS_ON_FLEXRAY	EcucEnumerationLiteralDef	
BSW Description	UDS on FlexRay (Manufacturer specific; ISO14229-1)	
Template Description	<p><b>DiagnosticConnection.physicalRequest:</b> Reference to a physical request message.</p> <p><b>FlexrayTpConnection:</b> A connection identifies the sender and the receiver of this particular communication. The FlexRayTp module routes a Pdu through this connection.</p> <p>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.</p>	
M2 Parameter		

SystemTemplate::DiagnosticConnection::DiagnosticConnection.physicalRequest, SystemTemplate::TransportProtocols::FlexrayTpConnection	
<b>Mapping Rule</b>	<b>Mapping Type</b>
In case DiagnosticConnection.physicalRequest exists and TpConnection Ident.ident belongs to FlexRayTpConnection	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00066

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolType
<b>BSW Parameter</b>	<b>BSW Type</b>
DCM_UDS_ON_IP	EcucEnumerationLiteralDef
<b>BSW Description</b>	
<b>Template Description</b>	
<p><b>DiagnosticConnection.physicalRequest:</b> Reference to a physical request message.</p> <p><b>SocketConnection:</b> The SoAd serves as a (De)Multiplexer between different PDU sources and the TCP/IP stack.</p>	
<b>M2 Parameter</b>	
SystemTemplate::DiagnosticConnection::DiagnosticConnection.physicalRequest, SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection	
<b>Mapping Rule</b>	<b>Mapping Type</b>
In case DiagnosticConnection.physicalRequest exists and TpConnection Ident.ident belongs to a SocketConnection	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00067

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspComControl/DcmDspComControlAllChannel
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspAllComMChannelRef	EcucReferenceDef
<b>BSW Description</b>	
Reference to ComM channel.	
<b>Template Description</b>	
<p>This reference represents the semantics that all available channels shall be affected. It is still necessary to refer to individual CommunicationClusters because there could be private CommunicationClusters in the System Extract that are not subject to the service "communication control".</p> <p>By referring to the applicable CommunicationClusters it can be made sure that only the affected CommunicationClusters are accessed.</p>	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl::DiagnosticComControlClass.allChannels	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00072

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspComControl/DcmDspComControlSpecificChannel	
BSW Parameter		BSW Type
DcmDspSpecificComMChannelRef		EcucReferenceDef
BSW Description		
Reference to ComM channel.		
Template Description		
This represents the ability to add additional attributes to the case that only specific channels are supposed to be considered,		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl::DiagnosticComControlClass.specificChannel		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00073

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspComControl/DcmDspComControlSpecificChannel	
BSW Parameter		BSW Type
DcmDspSubnetNumber		EcucIntegerParamDef
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		
DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl::DiagnosticComControlSpecificChannel.subnetNumber		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00074

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspComControl	
BSW Parameter		BSW Type
DcmDspComControlSubNode		EcucParamConfContainerDef
BSW Description		
This container gives information about the node identification number and the ComM channel used to address a sub-network.		
Template Description		
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".		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl::DiagnosticComControlClass.subNodeChannel		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00075

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspComControl/DcmDspComControlSubNode	
BSW Parameter		BSW Type
DcmDspComControlSubNodeComMChannelRef		EcucReferenceDef
BSW Description		
This parameter references a ComM channel where this node is connected to.		
Template Description		
This represents the affected CommunicationClusters in the role subNodeChannel		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl::DiagnosticComControlSubNodeChannel.subNodeChannel		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00076

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspComControl/DcmDspComControlSubNode	
BSW Parameter		BSW Type
DcmDspComControlSubNodeId		EcucIntegerParamDef
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 nodeIdentificationNumber of diagnostic service CommunicationControl (0x28).		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl::DiagnosticComControlSubNodeChannel.subNodeNumber		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00077

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp	
BSW Parameter		BSW Type
DcmDspCommonAuthorization		EcucParamConfContainerDef
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		
DiagnosticExtract::Dcm::DiagnosticService::RoutineControl::DiagnosticRoutineControl		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00254

BSW Module		BSW Context	
Dcm		Dcm/DcmConfigSet/DcmDsp/DcmDspControlDTCSetting	
BSW Parameter		BSW Type	
DcmSupportDTCSettingControlOptionRecord		EcucBooleanParamDef	
BSW Description			
This configuration switch defines if the DTCSettingControlOptionRecord is in general supported in the request message or not.			
Template Description			
This represents the decision whether the DTCSettingControlOptionRecord (see ISO 14229-1) is in general supported in the request message.			
M2 Parameter			
DiagnosticExtract::Dcm::DiagnosticService::ControlDTCSetting::DiagnosticControlDTCSettingClass.controlOptionRecordPresent			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		Mapping ID	
valid			

BSW Module		BSW Context	
Dcm		Dcm/DcmConfigSet/DcmDsp	
BSW Parameter		BSW Type	
DcmDspDDDIDcheckPerSourceDID		EcucBooleanParamDef	
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 ReadDataByIdentifier (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 ReadDataByIdentifier (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 ReadDataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF.			
M2 Parameter			
DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier::DiagnosticDynamicallyDefineDataIdentifierClass.checkPerSourceId			
Mapping Rule		Mapping Type	
1:1 mapping		full	
Mapping Status		Mapping ID	
valid		up_Dcm_00079	

BSW Module		BSW Context	
Dcm		Dcm/DcmConfigSet/DcmDsp/DcmDspData	
BSW Parameter		BSW Type	
DcmDspDataByteSize		EcucIntegerParamDef	
BSW Description			
Defines the array length in bytes or the the maximum array length for variable datalengths.			
Template Description			

<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
S/R via array: DcmDspDataByteSize= maxNumberOfElements * (baseTypeSize / 8)  C/S of FNC callback: DcmDspDataByteSize= maxNumberOfElements Note: 8 is the baseTypeSize of UINT8	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00085

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspDataConditionCheckReadFnc	EcucFunctionNameDef
<b>BSW Description</b>	
Function name to demand application if the conditions (e.g. System state) to read the DID are correct. (ConditionCheckRead-function).  Multiplicity shall be equal to parameter DcmDspDataReadFnc. This parameter is related to the interface Xxx_ConditionCheckRead.	
<b>Template Description</b>	
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.	
<b>M2 Parameter</b>	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency	
<b>Mapping Rule</b>	<b>Mapping Type</b>
It could be possible to get the FNC name via BswServiceDependency	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00080

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspDataEndianness	EcucEnumerationParamDef
<b>BSW Description</b>	
Defines the endianness of the data belonging to a DID in a diagnostic request or response message.	
<b>Template Description</b>	
This attribute specifies the byte order of the base type.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder	
<b>Mapping Rule</b>	<b>Mapping Type</b>



baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticDataIdentifier.dataElement	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00081

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	BSW Type
DcmDspDataFreezeCurrentStateFnc	EcucFunctionNameDef
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	
<b>DiagnosticServiceSwMapping.mappedBswServiceDependency:</b> 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.	
<b>DiagnosticIoControlNeeds.freezeCurrentStateSupported:</b> This attribute determines, if the referenced port supports temporary freezing of I/O value.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency, CommonStructure::ServiceNeeds::DiagnosticIoControlNeeds.freezeCurrentStateSupported	
Mapping Rule	Mapping Type
It could be possible to get the FNC name via BswServiceDependency	full
Mapping Status	Mapping ID
valid	up_Dcm_00004

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	BSW Type
DcmDspDataReadDataLengthFnc	EcucFunctionNameDef
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	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	Mapping Type
It could be possible to get the FNC name via BswServiceDependency	full
Mapping Status	Mapping ID
valid	up_Dcm_00082

BSW Module	BSW Context
------------	-------------

Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspDataReadFnc		EcucFunctionNameDef
<b>BSW Description</b>		
Function name to request from application the data value of a DID. (ReadData-function). This parameter is related to the interface Xxx_ReadData.		
<b>Template Description</b>		
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.		
<b>M2 Parameter</b>		
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency		
<b>Mapping Rule</b>		<b>Mapping Type</b>
It could be possible to get the FNC name via BswServiceDependency		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00083

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspDataResetToDefaultFnc		EcucFunctionNameDef
<b>BSW Description</b>		
Function name to request to application to reset an IOControl to default value. (ResetToDefault-function). This parameter is related to the interface Xxx_ResetToDefault.		
<b>Template Description</b>		
<b>DiagnosticServiceSwMapping.mappedBswServiceDependency:</b> 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.		
<b>DiagnosticIoControlNeeds.resetToDefaultSupported:</b> This represents a flag for the existence of the ResetToDefault operation in the service interface.		
<b>M2 Parameter</b>		
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency, CommonStructure::ServiceNeeds::DiagnosticIoControlNeeds.resetToDefaultSupported		
<b>Mapping Rule</b>		<b>Mapping Type</b>
It could be possible to get the FNC name via BswServiceDependency		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00005

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspDataReturnControlToEcuFnc		EcucFunctionNameDef
<b>BSW Description</b>		
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.		

<b>Template Description</b>	
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.	
<b>M2 Parameter</b>	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency	
<b>Mapping Rule</b>	<b>Mapping Type</b>
It could be possible to get the FNC name via BswServiceDependency	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00084

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspDataShortTermAdjustmentFnc	EcucFunctionNameDef
<b>BSW Description</b>	
Function name to request to application to adjust the IO signal. (ShortTermAdjustment-function).	
This parameter is related to the interface Xxx_ShortTermAdjustment.	
<b>Template Description</b>	
<b>DiagnosticServiceSwMapping.mappedBswServiceDependency:</b> 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.	
<b>DiagnosticIoControlNeeds.shortTermAdjustmentSupported:</b> This attribute determines, if the referenced port supports temporarily setting of I/O value to a specific value provided by the diagnostic tester.	
<b>M2 Parameter</b>	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency, CommonStructure::ServiceNeeds::DiagnosticIoControlNeeds.shortTermAdjustmentSupported	
<b>Mapping Rule</b>	<b>Mapping Type</b>
It could be possible to get the FNC name via BswServiceDependency	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00006

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
<b>BSW Parameter</b>	<b>BSW Type</b>
BOOLEAN	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the data is boolean.	
<b>Template Description</b>	

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticValueNeeds.fixedLength:</b> This attribute controls whether the data length of the data is fixed.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	
<b>Mapping Rule</b>	<b>Mapping Type</b>
baseTypeEncoding = BOOLEAN baseTypeSize = 1 maxNumberOfElements shall not exist arraySizeSemantics shall not exist  Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00008

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
<b>BSW Parameter</b>	<b>BSW Type</b>
SINT16	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the data is sint16.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>DiagnosticValueNeeds.fixedLength:</b> This attribute controls whether the data length of the data is fixed.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	
<b>Mapping Rule</b>	<b>Mapping Type</b>
baseTypeEncoding = 2C baseTypeSize = 16 maxNumberOfElements shall not exist arraySizeSemantics shall not exist  Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00012

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType	
BSW Parameter	BSW Type	
SINT16_N	EcucEnumerationLiteralDef	
BSW Description		
Type of the data is sint16 array.		
Template Description		
<p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p> <p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.</p> <p><b>DiagnosticDataElement.maxNumberOfElements:</b> 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.</p> <p><b>DiagnosticValueNeeds.fixedLength:</b> This attribute controls whether the data length of the data is fixed.</p>		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength		
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 ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)  Derivation from DiagnosticValueNeeds.fixedLength=1 possible.		full
Mapping Status		Mapping ID
valid		up_Dcm_00018

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType	
BSW Parameter	BSW Type	
SINT32	EcucEnumerationLiteralDef	
BSW Description		
Type of the data is sint32.		
Template Description		

<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>DiagnosticValueNeeds.fixedLength:</b> This attribute controls whether the data length of the data is fixed.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	
<b>Mapping Rule</b>	<b>Mapping Type</b>
baseTypeEncoding = 2C baseTypeSize = 32 maxNumberOfElements shall not exist arraySizeSemantics shall not exist  Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00014

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
<b>BSW Parameter</b>	<b>BSW Type</b>
SINT32_N	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the data is sint32 array.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>DiagnosticValueNeeds.fixedLength:</b> This attribute controls whether the data length of the data is fixed.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	
<b>Mapping Rule</b>	<b>Mapping Type</b>

<p>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)</p> <p>Derivation from DiagnosticValueNeeds.fixedLength=1 possible.</p>	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00020

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	BSW Type
SINT8	EcucEnumerationLiteralDef
BSW Description	
Type of the data is sint8.	
Template Description	
<p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p> <p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>DiagnosticValueNeeds.fixedLength:</b> This attribute controls whether the data length of the data is fixed.</p>	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	
Mapping Rule	Mapping Type
<p>baseTypeEncoding = 2C baseTypeSize = 8 maxNumberOfElements shall not exist arraySizeSemantics shall not exist</p> <p>Derivation from DiagnosticValueNeeds.fixedLength=1 possible.</p>	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00010

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	BSW Type
SINT8_N	EcucEnumerationLiteralDef
BSW Description	
Type of the data is sint8 array.	
Template Description	

<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>DiagnosticValueNeeds.fixedLength:</b> This attribute controls whether the data length of the data is fixed.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	
<b>Mapping Rule</b>	<b>Mapping Type</b>
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.fixedSize (cf. TPS_DEXT_01001)  Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00016

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT16	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the data is uint16.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticValueNeeds.fixedLength:</b> This attribute controls whether the data length of the data is fixed.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	



Mapping Rule	Mapping Type
baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16 maxNumberOfElements shall not exist arraySizeSemantics shall not exist  Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
Mapping Status	Mapping ID
valid	up_Dcm_00011

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	BSW Type
UINT16_N	EcucEnumerationLiteralDef
BSW Description	Type of the data is uint16 array.
Template Description	<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p> <p><b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.</p> <p><b>DiagnosticDataElement.maxNumberOfElements:</b> 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.</p> <p><b>DiagnosticValueNeeds.fixedLength:</b> This attribute controls whether the data length of the data is fixed.</p>
M2 Parameter	AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength
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 ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)  Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
Mapping Status	Mapping ID
valid	up_Dcm_00017

BSW Module	BSW Context
------------	-------------

Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType	
<b>BSW Parameter</b>		<b>BSW Type</b>
UINT32		EcucEnumerationLiteralDef
<b>BSW Description</b>		
Type of the data is uint32.		
<b>Template Description</b>		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>DiagnosticValueNeeds.fixedLength:</b> This attribute controls whether the data length of the data is fixed.		
<b>M2 Parameter</b>		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength		
<b>Mapping Rule</b>		<b>Mapping Type</b>
baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32 maxNumberOfElements shall not exist arraySizeSemantics shall not exist  Derivation from DiagnosticValueNeeds.fixedLength=1 possible.		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00013

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType	
<b>BSW Parameter</b>		<b>BSW Type</b>
UINT32_N		EcucEnumerationLiteralDef
<b>BSW Description</b>		
Type of the data is uint32 array.		
<b>Template Description</b>		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.		
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.		
<b>DiagnosticValueNeeds.fixedLength:</b> This attribute controls whether the data length of the data is fixed.		
<b>M2 Parameter</b>		

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	
<b>Mapping Rule</b>	<b>Mapping Type</b>
baseTypeEncoding = NONE, UTF-32 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)  Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00019

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT8	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the data is uint8.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticValueNeeds.fixedLength:</b> This attribute controls whether the data length of the data is fixed.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	
<b>Mapping Rule</b>	<b>Mapping Type</b>
baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8 maxNumberOfElements shall not exist arraySizeSemantics shall not exist  Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00009

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT8_DYN	EcucEnumerationLiteralDef
<b>BSW Description</b>	

Type of the data is uint8 array with dynamic length.	
<b>Template Description</b>	
<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p> <p><b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.</p> <p><b>DiagnosticDataElement.maxNumberOfElements:</b> 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.</p> <p><b>DiagnosticValueNeeds.fixedLength:</b> This attribute controls whether the data length of the data is fixed.</p>	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	
<b>Mapping Rule</b>	<b>Mapping Type</b>
baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP 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) Derivation from DiagnosticValueNeeds.fixedLength=0 possible.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00007

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT8_N	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the data is uint8 array.	
<b>Template Description</b>	

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>DiagnosticValueNeeds.fixedLength:</b> This attribute controls whether the data length of the data is fixed.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	
<b>Mapping Rule</b>	<b>Mapping Type</b>
baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP 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.fixedSize (cf. TPS_DEXT_01001)  Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00015

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspDataUsePort		EcucEnumerationParamDef
<b>BSW Description</b>		
Defines which interface shall be used to access the data.		
<b>Template Description</b>		
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.		
<b>M2 Parameter</b>		
CommonStructure::ServiceNeeds::DiagnosticValueNeeds.processingStyle		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
1:1 mapping	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid	up_Dcm_00001	

<b>BSW Module</b>	<b>BSW Context</b>
-------------------	--------------------

Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort	
<b>BSW Parameter</b>		<b>BSW Type</b>
USE_DATA_ASYNC_CLIENT_SERVER		EcucEnumerationLiteralDef
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
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.		
<b>M2 Parameter</b>		
CommonStructure::ServiceNeeds::DiagnosticProcessingStyleEnum.processingStyleAsynchronous		
<b>Mapping Rule</b>		<b>Mapping Type</b>
DiagnosticServiceSwMapping is having a SwcServiceDependency and ServiceNeeds::DiagnosticProcessingStyleEnum is equal to processingStyleAsynchronous		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00022

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort	
<b>BSW Parameter</b>		<b>BSW Type</b>
USE_DATA_ASYNC_CLIENT_SERVER_ERROR		EcucEnumerationLiteralDef
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
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.		
<b>M2 Parameter</b>		
CommonStructure::ServiceNeeds::DiagnosticProcessingStyleEnum.processingStyleAsynchronousWithError		
<b>Mapping Rule</b>		<b>Mapping Type</b>
DiagnosticServiceSwMapping is having a SwcServiceDependency and ServiceNeeds::DiagnosticProcessingStyleEnum is equal to processingStyleAsynchronousWithError		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00023

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort	
<b>BSW Parameter</b>		<b>BSW Type</b>
USE_DATA_ASYNC_FNC		EcucEnumerationLiteralDef
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
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.		

<b>M2 Parameter</b>	
CommonStructure::ServiceNeeds::DiagnosticProcessingStyleEnum.processingStyleAsynchronous	
<b>Mapping Rule</b>	<b>Mapping Type</b>
DiagnosticServiceSwMapping is having a BswServiceDependency and ServiceNeeds::DiagnosticProcessingStyleEnum is equal to processingStyleAsynchronous	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00250

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort
<b>BSW Parameter</b>	<b>BSW Type</b>
USE_DATA_SENDER_RECEIVER	EcucEnumerationLiteralDef
<b>BSW Description</b>	
The DCM will access the Data using an Port requiring a SenderReceiverInteface (with isService=false) DataServices_{Data}. The Port is namedDataServices_{Data} where {Data} is the name of the container DcmDspData.	
<b>Template Description</b>	
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.	
<b>M2 Parameter</b>	
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00087

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort
<b>BSW Parameter</b>	<b>BSW Type</b>
USE_DATA_SENDER_RECEIVER_AS_SERVICE	EcucEnumerationLiteralDef
<b>BSW Description</b>	
The DCM will access the Data using an service Port requiring a SenderReceiverInteface (with isService=true) DataServices_{Data} . The Port is namedDataServices_{Data} where {Data} is the name of the container DcmDspData.	
<b>Template Description</b>	
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.	
<b>M2 Parameter</b>	
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00088

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData

BSW Parameter		BSW Type
DcmDspDataWriteFnc		EcucFunctionNameDef
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		
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency		
Mapping Rule		Mapping Type
It could be possible to get the FNC name via BswServiceDependency		full
Mapping Status		Mapping ID
valid		up_Dcm_00090

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDiagnosisScaling/DcmDspAlternativeDataType	
BSW Parameter		BSW Type
DcmDspTextTableMapping		EcucParamConfContainerDef
BSW Description		
<p>The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists).</p> <p>In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.</p>		
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		
AsamHdo::ComputationMethod::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		Mapping ID
valid		up_Dcm_00097

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDiagnosisScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping	
BSW Parameter		BSW Type
DcmDspDiagnosisRepresentationDataValue		EcucIntegerParamDef
BSW Description		



The data value in the diagnosis representation.	
<b>Template Description</b>	
This represents a textual constant in the computation method.	
<b>M2 Parameter</b>	
AsamHdo::ComputationMethod::CompuConstTextContent.vt	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00098

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDiagnosisScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspInternalDataValue		EcucIntegerParamDef
<b>BSW Description</b>		
The ECU internal data value.		
<b>Template Description</b>		
<b>CompuScale.lowerLimit:</b> This specifies the lower limit of the scale.		
<b>CompuScale.upperLimit:</b> This specifies the upper limit of a of the scale.		
<b>M2 Parameter</b>		
AsamHdo::ComputationMethod::CompuScale.lowerLimit, AsamHdo::ComputationMethod::CompuScale.upperLimit		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
1:1 mapping	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid	up_Dcm_00099	

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspDidDataSupportInfo		EcucParamConfContainerDef
<b>BSW Description</b>		
This container defines the supported information.		
<b>Template Description</b>		
<b>DiagnosticParameter.supportInfo:</b> This attribute represents the ability to define which bit of the support info byte is representing this part of the PID.		
<b>DiagnosticDataIdentifier.supportInfoByte:</b> This attribute represents the supported information associated with the DiagnosticDataIdentifier.		
<b>M2 Parameter</b>		
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.supportInfo, DiagnosticExtract::CommonDiagnostics::DiagnosticDataIdentifier.supportInfoByte		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
1:1 mapping	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid	up_Dcm_00282	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspExternalSRDataElement Class/DcmDataElementInstance	
BSW Parameter		BSW Type
DcmDataElementInstanceRef		EcucInstanceReferenceDef
BSW Description		
<p>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</p>		
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		
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping.mappedDataElement		
Mapping Rule		Mapping Type
DiagnosticServiceDataMapping maps to a primitive data.		full
Mapping Status		Mapping ID
valid		up_Dcm_00100

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspExternalSRDataElement Class/DcmSubElementInDataElementInstance	
BSW Parameter		BSW Type
DcmSubElementInDataElementInstanceRef		EcucInstanceReferenceDef
BSW Description		
<p>Instance Reference to the primitive sub-element (at any level) of composite data in a port which shall be read. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationCompositeDataType.</p>		
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		
DiagnosticExtract::ServiceMapping::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		Mapping ID
valid		up_Dcm_00101

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspExternalSRDataElement Class/DcmSubElementInImplDataElementInstance	
BSW Parameter		BSW Type
DcmSubElementInImplDataElementInstanceRef		EcucInstanceReferenceDef

<b>BSW Description</b>	
Instance Reference to the primitive sub-element (at any level) of composite data in a port which shall be read. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ImplementationDataType of category STRUCTURE or ARRAY. Please note that in case of ARRAY the index attribute in the target reference has to be set to select a single array element.	
<b>Template Description</b>	
This represents the dataElement in the application software that is accessed for diagnostic purpose. This role is applicable on the classic platform.	
<b>M2 Parameter</b>	
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping.mappedDataElement	
<b>Mapping Rule</b>	<b>Mapping Type</b>
DiagnosticServiceDataMapping maps to a primitive element within a composite data, where the AutosarDataPrototype is typed with a ApplicationCompositeDataType ImplementationDataType of category STRUCTURE or ARRAY.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00102

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspOdxDataDescription		EcucAddInfoParamDef
<b>BSW Description</b>		
Defines additional description for ODX documentation		
<b>Template Description</b>		
This specifies the long name of the object. Long name is targeted to human readers and acts like a headline.		
<b>M2 Parameter</b>		
GenericStructure::GeneralTemplateClasses::Identifiable::MultilanguageReferrable.longName		
<b>Mapping Rule</b>		<b>Mapping Type</b>
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
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00103

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspDataDefaultEndianness		EcucEnumerationParamDef
<b>BSW Description</b>		
Defines the default endianness belonging to a DID, RID or PID if the corresponding data does not define an endianness.		
<b>Template Description</b>		
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.		
<b>M2 Parameter</b>		
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.defaultEndianness		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full

Mapping Status	Mapping ID
valid	up_Dcm_00104

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp	
BSW Parameter	BSW Type	
DcmDspDid	EcucParamConfContainerDef	
BSW Description		
This container contains the configuration (parameters) of the DID.		
Template Description		
<b>DiagnosticDataIdentifier:</b> This meta-class represents the ability to model a diagnostic data identifier (DID) that is fully specified regarding the payload at configuration-time.		
<b>DiagnosticDynamicDataIdentifier:</b> This meta-class represents the ability to define a diagnostic data identifier (DID) at run-time.		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticDataIdentifier DiagnosticExtract::CommonDiagnostics::DiagnosticDynamicDataIdentifier		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00178	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid	
BSW Parameter	BSW Type	
DcmDspDidIdentifier	EcucIntegerParamDef	
BSW Description		
2 byte Identifier of the DID		
Within each DcmConfigSet all DcmDspDidIdentifier values shall be unique.		
Template Description		
<b>DiagnosticAbstractDataIdentifier.id:</b> This is the numerical identifier used to identify the DiagnosticAbstractDataIdentifier in the scope of diagnostic workflow		
<b>DiagnosticValueNeeds.didNumber:</b> This represents a Data identifier for the diagnostic value. This allows to predefine the DID number if the responsible function developer has received a particular requirement from the OEM or from a standardization body.		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticAbstractDataIdentifier.id, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.didNumber		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00002	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidSignal

BSW Parameter		BSW Type
DcmDspDidByteOffset		EcucIntegerParamDef
BSW Description		
Defines the absolute byte offset of the data defined by DcmDspDidDataRef reference to DcmDsp-Data container in the DID.		
Template Description		
This represents the bitOffset of the DiagnosticParameter		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.bitOffset		
Mapping Rule		Mapping Type
bitOffset / 8		full
Mapping Status		Mapping ID
valid		up_Dcm_00283

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid	
BSW Parameter		BSW Type
DcmDspDidSize		EcucIntegerParamDef
BSW Description		
Length of a DID in byte(s).		
Template Description		
This attribute indicates the size in bytes of the DiagnosticDataIdentifier.		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticDataIdentifier.didSize		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00280

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid	
BSW Parameter		BSW Type
DcmDspDidSupportInfo		EcucParamConfContainerDef
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		
DiagnosticExtract::CommonDiagnostics::DiagnosticDataIdentifier.supportInfoByte		
Mapping Rule		Mapping Type
		full
Mapping Status		Mapping ID
valid		up_Dcm_00281

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo	
BSW Parameter		BSW Type
DcmDspDDDIDMaxElements		EcucIntegerParamDef
BSW Description		

Maximum number of source elements of a DDDID.	
<b>Template Description</b>	
This represents the maximum number of source elements of the dynamically created DID.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier::DiagnosticDynamicallyDefineDataIdentifier.maxSourceElement	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00107

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspDidControlSecurityLevelRef		EcucReferenceDef
<b>BSW Description</b>		
Reference to DcmDspSecurityRow Security levels allowed to control this DID. If there is no reference, no check of security level shall be done.		
<b>Template Description</b>		
This represents the associated DiagnosticSecurityLevels		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticAccessPermission.securityLevel		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00108

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspDidControlSessionRef		EcucReferenceDef
<b>BSW Description</b>		
Reference to DcmDspSessionRow Sessions allowed to control this DID. If there is no reference, no check of session level shall be done.		
<b>Template Description</b>		
This represents the associated DiagnosticSessions		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00109

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspDidFreezeCurrentState		EcucBooleanParamDef
<b>BSW Description</b>		
This indicates the presence of "FreezeCurrentState".		

Template Description	
<b>DiagnosticIOControl.freezeCurrentState:</b> Setting this attribute to true represents the ability of the Dcm to execute a freezeCurrentState.	
<b>DiagnosticIOControlNeeds.freezeCurrentStateSupported:</b> This attribute determines, if the referenced port supports temporary freezing of I/O value.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::IOControl::DiagnosticIOControl.freezeCurrentState, CommonStructure::ServiceNeeds::DiagnosticIOControlNeeds.freezeCurrentStateSupported	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	Mapping ID
valid	up_Dcm_00035

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl
BSW Parameter	BSW Type
DcmDspDidResetToDefault	EcucBooleanParamDef
BSW Description	
This indicates the presence of "ResetToDefault".	
Template Description	
<b>DiagnosticIOControl.resetToDefault:</b> Setting this attribute to true represents the ability of the Dcm to execute a resetToDefault.	
<b>DiagnosticIOControlNeeds.resetToDefaultSupported:</b> This represents a flag for the existence of the ResetToDefault operation in the service interface.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::IOControl::DiagnosticIOControl.resetToDefault, CommonStructure::ServiceNeeds::DiagnosticIOControlNeeds.resetToDefaultSupported	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	Mapping ID
valid	up_Dcm_00036

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl
BSW Parameter	BSW Type
DcmDspDidShortTermAdjustment	EcucBooleanParamDef
BSW Description	
This indicates the presence of "ShortTermAdjustment".	
Template Description	
<b>DiagnosticIOControl.shortTermAdjustment:</b> Setting this attribute to true represents the ability of the Dcm to execute a shortTermAdjustment.	
<b>DiagnosticIOControlNeeds.shortTermAdjustmentSupported:</b> 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	
DiagnosticExtract::Dcm::DiagnosticService::IOControl::DiagnosticIOControl.shortTermAdjustment, CommonStructure::ServiceNeeds::DiagnosticIOControlNeeds.shortTermAdjustmentSupported	
Mapping Rule	Mapping Type
1:1 mapping	full

Mapping Status	Mapping ID
valid	up_Dcm_00037

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo
BSW Parameter	BSW Type
DcmDspDidDynamicallyDefined	EcucBooleanParamDef
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	
<b>DiagnosticDataIdentifier:</b> This meta-class represents the ability to model a diagnostic data identifier (DID) that is fully specified regarding the payload at configuration-time.	
<b>DiagnosticDynamicDataIdentifier:</b> This meta-class represents the ability to define a diagnostic data identifier (DID) at run-time.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticDataIdentifier, DiagnosticExtract::CommonDiagnostics::DiagnosticDynamicDataIdentifier	
Mapping Rule	Mapping Type
true: in case the DiagnosticAbstractDataIdentifier for the DID value is aggregated by DiagnosticDynamicDataIdentifier false: in case the DiagnosticAbstractDataIdentifier for the DID value is aggregated by DiagnosticDataIdentifier	full
Mapping Status	Mapping ID
valid	up_Dcm_00110

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidRead
BSW Parameter	BSW Type
DcmDspDidReadSecurityLevelRef	EcucReferenceDef
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	
DiagnosticExtract::Dcm::DiagnosticAccessPermission.securityLevel	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	Mapping ID
valid	up_Dcm_00111

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidRead
BSW Parameter	BSW Type
DcmDspDidReadSessionRef	EcucReferenceDef
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.	
<b>Template Description</b>	
This represents the associated DiagnosticSessions	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00112

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidWrite
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspDidWriteSecurityLevelRef	EcucReferenceDef
<b>BSW Description</b>	
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.	
<b>Template Description</b>	
This represents the associated DiagnosticSecurityLevels	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticAccessPermission.securityLevel	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00113

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidWrite
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspDidWriteSessionRef	EcucReferenceDef
<b>BSW Description</b>	
Reference to DcmDspSessionRow Referenced sessions are allowed to write this DID.	
If there is no reference, no check of session level shall be done.	
<b>Template Description</b>	
This represents the associated DiagnosticSessions	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00114

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspEcuReset/DcmDspEcuResetRow
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspEcuResetId	EcucIntegerParamDef

<b>BSW Description</b>	
Defines the subfunction ID	
<b>Template Description</b>	
This represents the maximum number of source elements of the dynamically created DID.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier::DiagnosticDynamicallyDefineDataIdentifier.maxSourceElement	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00107

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspEcuReset/DcmDspEcuResetRow
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmResponseToEcuReset	EcucEnumerationParamDef
<b>BSW Description</b>	
Defines the answer to EcuReset service should come: Before or after the reset.	
<b>Template Description</b>	
This attribute defines whether the response to the EcuReset service shall be transmitted before or after the actual reset.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::EcuReset::DiagnosticEcuResetClass.respondToReset	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00244

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspEcuReset/DcmDspEcuResetRow/DcmResponseToEcuReset
<b>BSW Parameter</b>	<b>BSW Type</b>
AFTER_RESET	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Answer to EcuReset service should come after the reset.	
<b>Template Description</b>	
Answer to EcuReset service should come after the reset.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::EcuReset::DiagnosticResponseToEcuResetEnum.respondAfterReset	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00245

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspEcuReset/DcmDspEcuResetRow/DcmResponseToEcuReset
<b>BSW Parameter</b>	<b>BSW Type</b>
BEFORE_RESET	EcucEnumerationLiteralDef

<b>BSW Description</b>	
Answer to EcuReset service should come before the reset.	
<b>Template Description</b>	
Answer to EcuReset service should come before the reset.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::EcuReset::DiagnosticResponseToEcuReset Enum.respondBeforeReset	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00246

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspMaxDidToRead		EcucIntegerParamDef
<b>BSW Description</b>		
Indicates the maximum allowed DID's in a single "ReadDataByIdentifier" request.		
<b>Template Description</b>		
This attribute represents the maximum number of allowed DID's in a single instance of DiagnosticReadDataByIdentifier.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier::DiagnosticReadDataByIdentifier Class.maxDidToRead		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
1:1 mapping	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid	up_Dcm_00115	

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspMaxPeriodicDidToRead		EcucIntegerParamDef
<b>BSW Description</b>		
Indicates the maximum allowed periodicDID's which can be read in a single "ReadDataByPeriodicIdentifier" request.		
<b>Template Description</b>		
This represents the maximum number of data identifiers that can be included in one request.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID::DiagnosticReadDataByPeriodicIDClass.maxPeriodicDidToRead		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
1:1 mapping	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid	up_Dcm_00116	

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspMemoryIdValue		EcucIntegerParamDef

<b>BSW Description</b>	
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.	
<b>Template Description</b>	
This represents the identification of the memory segment.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.id	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00117

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeByLabelInfo	
<b>BSW Parameter</b>	<b>BSW Type</b>	
DcmDspReadMemoryRangeByLabelHigh	EcucStringParamDef	
<b>BSW Description</b>		
High memory address as label (string) of a range allowed for reading.		
<b>Template Description</b>		
This represents a symbolic label for the upper bound for addresses of the memory segment.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.memoryHighAddressLabel		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
1:1 mapping	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid	up_Dcm_00118	

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeByLabelInfo	
<b>BSW Parameter</b>	<b>BSW Type</b>	
DcmDspReadMemoryRangeByLabelLow	EcucStringParamDef	
<b>BSW Description</b>		
Low memory address as label (string) of a range allowed for reading.		
<b>Template Description</b>		
This represents a symbolic label for the lower bound for addresses of the memory segment.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.memoryLowAddressLabel		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
1:1 mapping	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid	up_Dcm_00119	

<b>BSW Module</b>	<b>BSW Context</b>
-------------------	--------------------

Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeByLabelInfo	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspReadMemoryRangeSecurityLevelRef		EcucReferenceDef
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
This represents the associated DiagnosticSecurityLevels		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticAccessPermission.securityLevel		
<b>Mapping Rule</b>		<b>Mapping Type</b>
DiagnosticMemoryIdentifier referenced in the role memoryRange is referenced by a DiagnosticReadMemoryByAddress The accessPermission holds the security level information.		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00120

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeByLabelInfo	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspReadMemoryRangeSessionLevelRef		EcucReferenceDef
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
This represents the associated DiagnosticSessions		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00278

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeInfo	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspReadMemoryRangeHigh		EcucIntegerParamDef
<b>BSW Description</b>		
High memory address of a range allowed for reading		
<b>Template Description</b>		
This represents the upper bound for addresses of the memory segment.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.memoryHighAddress		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00121

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeInfo	
BSW Parameter		BSW Type
DcmDspReadMemoryRangeLow		EcucIntegerParamDef
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		
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.memoryLowAddress		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00122

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeInfo	
BSW Parameter		BSW Type
DcmDspReadMemoryRangeSecurityLevelRef		EcucReferenceDef
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		
DiagnosticExtract::Dcm::DiagnosticAccessPermission.securityLevel		
Mapping Rule		Mapping Type
DiagnosticMemoryIdentifier referenced in the role memoryRange is referenced by a DiagnosticReadMemoryByAddress The accessPermission holds the security level information.		full
Mapping Status		Mapping ID
valid		up_Dcm_00179

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeInfo	
BSW Parameter		BSW Type
DcmDspReadMemoryRangeSessionLevelRef		EcucReferenceDef
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		
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession		
Mapping Rule		Mapping Type
1:1 mapping		full

Mapping Status	Mapping ID
valid	up_Dcm_00276

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeByLabelInfo	
BSW Parameter	BSW Type	
DcmDspWriteMemoryRangeByLabelHigh	EcucStringParamDef	
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		
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.memoryHighAddressLabel		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00123	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeByLabelInfo	
BSW Parameter	BSW Type	
DcmDspWriteMemoryRangeByLabelLow	EcucStringParamDef	
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		
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.memoryLowAddressLabel		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00124	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeByLabelInfo	
BSW Parameter	BSW Type	
DcmDspWriteMemoryRangeSecurityLevelRef	EcucReferenceDef	
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		
DiagnosticExtract::Dcm::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
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00125

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeByLabelInfo	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspWriteMemoryRangeSessionLevelRef		EcucReferenceDef
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
This represents the associated DiagnosticSessions		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00279

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeInfo	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspWriteMemoryRangeHigh		EcucIntegerParamDef
<b>BSW Description</b>		
High memory address of a range allowed for writing.		
<b>Template Description</b>		
This represents the upper bound for addresses of the memory segment.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.memoryHighAddress		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00126

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeInfo	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspWriteMemoryRangeLow		EcucIntegerParamDef
<b>BSW Description</b>		
Low memory address of a range allowed for writing		
<b>Template Description</b>		
This represents the lower bound for addresses of the memory segment.		



<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.memoryLowAddress	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00127

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeInfo	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspWriteMemoryRangeSecurityLevelRef		EcucReferenceDef
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
This represents the associated DiagnosticSecurityLevels		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticAccessPermission.securityLevel		
<b>Mapping Rule</b>		<b>Mapping Type</b>
DiagnosticMemoryIdentifier referenced in the role memoryRange is referenced by a DiagnosticWriteMemoryByAddress The accessPermission holds the security level information.		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00180

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeInfo	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspWriteMemoryRangeSessionLevelRef		EcucReferenceDef
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
This represents the associated DiagnosticSessions		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00277

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemoryTransfer/DcmDspMemoryTransferIdInfo	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspMemoryIdValue		EcucIntegerParamDef
<b>BSW Description</b>		

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.	
<b>Template Description</b>	
This represents the identification of the memory segment.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.id	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00117

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPeriodicTransmission	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspMaxPeriodicDidScheduler		EcucIntegerParamDef
<b>BSW Description</b>		
Defines the maximum number of periodicDataIdentifiers that can be scheduled concurrently.		
<b>Template Description</b>		
This represents the maximum number of periodic data identifiers that can be scheduled in parallel.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID::DiagnosticReadDataByPeriodicIDClass.schedulerMaxNumber		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00128

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPeriodicTransmission	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspPeriodicTransmissionFastRate		EcucFloatParamDef
<b>BSW Description</b>		
This parameter give the transmission rate of the requested periodicDataIdentifiers to be used if the parameter transmissionMode given in the ReadDataByPeriodicID request is equal to 0x03 ("sendAtFastRate"). This parameter value in seconds have to be configured as a multiple of DcmTaskTime.		
min: A negative value and zero is not allowed.		
<b>Template Description</b>		
<b>DiagnosticPeriodicRate.period:</b> This represents the period of the DiagnosticPeriodicRate in seconds.		
<b>DiagnosticPeriodicRate.periodicRateCategory:</b> This attribute represents the category of the periodic rate.		
<b>M2 Parameter</b>		

DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID::DiagnosticPeriodicRate.period DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID::DiagnosticPeriodicRate.periodicRateCategory	
<b>Mapping Rule</b>	<b>Mapping Type</b>
The parameter shall exist if DiagnosticPeriodicRate.periodicRateCategory is set to DiagnosticPeriodicRateCategoryEnum.periodicRateFast.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00129

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPeriodicTransmission
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspPeriodicTransmissionMediumRate	EcucFloatParamDef
<b>BSW Description</b>	
<p>This parameter give the transmission rate of the requested periodicDataIdentifiers to be used if the parameter transmissionMode given in the ReadDataByPeriodicID request is equal to 0x02 ("sendAtMediumRate"). This parameter value in seconds have to be configured as a multiple of DcmTaskTime.</p> <p>min: A negative value and zero is not allowed.</p>	
<b>Template Description</b>	
<p><b>DiagnosticPeriodicRate.period:</b> This represents the period of the DiagnosticPeriodicRate in seconds.</p> <p><b>DiagnosticPeriodicRate.periodicRateCategory:</b> This attribute represents the category of the periodic rate.</p>	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID::DiagnosticPeriodicRate.period DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID::DiagnosticPeriodicRate.periodicRateCategory	
<b>Mapping Rule</b>	<b>Mapping Type</b>
The parameter shall exist if DiagnosticPeriodicRate.periodicRateCategory is set to DiagnosticPeriodicRateCategoryEnum.periodicRateMedium.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00130

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPeriodicTransmission
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspPeriodicTransmissionSlowRate	EcucFloatParamDef
<b>BSW Description</b>	
<p>This parameter give the transmission rate of the requested periodicDataIdentifiers to be used if the parameter transmissionMode given in the ReadDataByPeriodicID request is equal to 0x01 ("sendAtSlowRate"). This parameter value in seconds have to be configured as a multiple of DcmTaskTime.</p> <p>min: A negative value and zero is not allowed.</p>	
<b>Template Description</b>	

<b>DiagnosticPeriodicRate.period:</b> This represents the period of the DiagnosticPeriodicRate in seconds.	
<b>DiagnosticPeriodicRate.periodicRateCategory:</b> This attribute represents the category of the periodic rate.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID::DiagnosticPeriodicRate.period DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID::DiagnosticPeriodicRate.periodicRateCategory	
<b>Mapping Rule</b>	<b>Mapping Type</b>
The parameter shall exist if DiagnosticPeriodicRate.periodicRateCategory is set to DiagnosticPeriodicRateCategoryEnum.periodicRateSlow.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00131

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspPid	EcucParamConfContainerDef
<b>BSW Description</b>	
This container defines the availability of a PID to the DCM.	
<b>Template Description</b>	
This meta-class represents the ability to model a diagnostic parameter identifier (PID) for the purpose of executing on-board diagnostics (OBD).	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameterIdentifier	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspPidData	EcucParamConfContainerDef
<b>BSW Description</b>	
This container defines the parameter for a Signal in the PID.	
<b>Template Description</b>	
This represents the data carried by the DiagnosticParameterIdentifier.	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameterIdentifier.dataElement	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspPidByteOffset	EcucIntegerParamDef

<b>BSW Description</b>	
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).	
<b>Template Description</b>	
This represents the bitOffset of the DiagnosticParameter	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.bitOffset	
<b>Mapping Rule</b>	<b>Mapping Type</b>
bitOffset / 8	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00284

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspPidDataByteSize	EcucIntegerParamDef
<b>BSW Description</b>	
Defines the array length in bytes or the the maximum array length for variable datalengths.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
S/R via array: $DcmDspPidDataByteSize = \maxNumberOfElements * (baseTypeSize / 8)$  C/S of FNC callback: $DcmDspPidDataByteSize = \maxNumberOfElements$ Note: 8 is the baseTypeSize of UINT8	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00285

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspPidDataSupportInfo	EcucParamConfContainerDef
<b>BSW Description</b>	
This container defines the supported information.	
<b>Template Description</b>	
This attribute represents the ability to define which bit of the support info byte is representing this part of the PID.	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.supportInfo	
<b>Mapping Rule</b>	<b>Mapping Type</b>
Applicable if the DiagnosticParameter is owned by a DiagnosticParameterIdentifier	full

Mapping Status	Mapping ID
valid	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidDataSupportInfo	
BSW Parameter		BSW Type
DcmDspPidDataSupportInfoBit		EcucIntegerParamDef
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		
DiagnosticExtract::CommonDiagnostics::DiagnosticParameterSupportInfo.supportInfoBit		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidDataSupportInfo	
BSW Parameter		BSW Type
DcmDspPidDataSupportInfoRef		EcucReferenceDef
BSW Description		
Reference to DcmDspPidSupportInfo		
Template Description		
This represents the supported information associated with the DiagnosticParameterIdentifier.		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticParameterIdentifier.supportInfoByte		
Mapping Rule		Mapping Type
Shall refer to the DiagnosticParameterIdentifier.supportInfoByte of the enclosing DiagnosticParameterIdentifier		full
Mapping Status		Mapping ID
valid		

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspDiagnosisScaling/DcmDspAlternativeDataType	
BSW Parameter		BSW Type
DcmDspTextTableMapping		EcucParamConfContainerDef
BSW Description		

<p>The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists).</p> <p>In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.</p>	
<b>Template Description</b>	
<p>This meta-class represents the ability to express the relationship between a physical value and the mathematical representation.</p> <p>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.</p>	
<b>M2 Parameter</b>	
AsamHdo::ComputationMethod::CompuMethod	
<b>Mapping Rule</b>	<b>Mapping Type</b>
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00097

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspDiagnosisScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspDiagnosisRepresentationDataValue		EcucIntegerParamDef
<b>BSW Description</b>		
The data value in the diagnosis representation.		
<b>Template Description</b>		
This represents a textual constant in the computation method.		
<b>M2 Parameter</b>		
AsamHdo::ComputationMethod::CompuConstTextContent.vt		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00098

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspDiagnosisScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspInternalDataValue		EcucIntegerParamDef
<b>BSW Description</b>		
The ECU internal data value.		
<b>Template Description</b>		

<b>CompuScale.lowerLimit:</b> This specifies the lower limit of the scale.	
<b>CompuScale.upperLimit:</b> This specifies the upper limit of a of the scale.	
<b>M2 Parameter</b>	
AsamHdo::ComputationMethod::CompuScale.lowerLimit, AsamHdo::ComputationMethod::CompuScale.upperLimit	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00099

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataEndianness	
<b>BSW Parameter</b>		<b>BSW Type</b>
BIG_ENDIAN		EcucEnumerationLiteralDef
<b>BSW Description</b>		
Most significant byte shall be stored at the lowest address.		
<b>Template Description</b>		
This attribute specifies the byte order of the base type.		
<b>M2 Parameter</b>		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder		
<b>Mapping Rule</b>		<b>Mapping Type</b>
BaseTypeDirectDefinition.byteOrder == ByteOrderEnum.mostSignificantByte First		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataEndianness	
<b>BSW Parameter</b>		<b>BSW Type</b>
LITTLE_ENDIAN		EcucEnumerationLiteralDef
<b>BSW Description</b>		
Most significant byte shall be stored at the highest address		
<b>Template Description</b>		
This attribute specifies the byte order of the base type.		
<b>M2 Parameter</b>		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder		
<b>Mapping Rule</b>		<b>Mapping Type</b>
BaseTypeDirectDefinition.byteOrder ==ByteOrderEnum.mostSignificantByte Last		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataEndianness	
<b>BSW Parameter</b>		<b>BSW Type</b>



OPAQUE	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Opaque data endianness	
<b>Template Description</b>	
This attribute specifies the byte order of the base type.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder	
<b>Mapping Rule</b>	<b>Mapping Type</b>
BaseTypeDirectDefinition.byteOrder == ByteOrderEnum.opaque	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01	
<b>BSW Parameter</b>	<b>BSW Type</b>	
DcmDspPidDataReadFnc	EcucFunctionNameDef	
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
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.		
<b>M2 Parameter</b>		
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
The BswServiceDependency should have aRoleBasedBswModuleEntryAssignment that in turn has attribute role set to xxx_ReadData and points to a BswModuleEntry.	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid		

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType	
<b>BSW Parameter</b>	<b>BSW Type</b>	
BOOLEAN	EcucEnumerationLiteralDef	
<b>BSW Description</b>		
Type of the data is boolean.		
<b>Template Description</b>		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>M2 Parameter</b>		

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement  baseTypeEncoding = BOOLEAN baseTypeSize = 1	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType	
<b>BSW Parameter</b>		<b>BSW Type</b>
SINT16		EcucEnumerationLiteralDef
<b>BSW Description</b>		
Type of the data is sint16.		
<b>Template Description</b>		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>M2 Parameter</b>		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
<b>Mapping Rule</b>		<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement  baseTypeEncoding = 2C baseTypeSize = 16		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00154

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType	
<b>BSW Parameter</b>		<b>BSW Type</b>
SINT16_N		EcucEnumerationLiteralDef
<b>BSW Description</b>		
Type of the data is sint16 array.		
<b>Template Description</b>		

<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements,	
<b>Mapping Rule</b>	<b>Mapping Type</b>
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 ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00160

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType
<b>BSW Parameter</b>	<b>BSW Type</b>
SINT32	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the data is sint32.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement  baseTypeEncoding = 2C baseTypeSize = 32	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType	
BSW Parameter		BSW Type
SINT32_N		EcucEnumerationLiteralDef
BSW Description		
Type of the data is sint32 array.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.		
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::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 ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)		full
Mapping Status		Mapping ID
valid		up_Dcm_00166

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType	
BSW Parameter		BSW Type
SINT8		EcucEnumerationLiteralDef
BSW Description		
Type of the data is sint8.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement  baseTypeEncoding = 2C baseTypeSize = 8	full
Mapping Status	Mapping ID
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType
BSW Parameter	BSW Type
SINT8_N	EcucEnumerationLiteralDef
BSW Description	Type of the data is sint8 array.
Template Description	<p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p> <p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.</p> <p><b>DiagnosticDataElement.maxNumberOfElements:</b> 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.</p>
M2 Parameter	AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::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.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	Mapping ID
valid	up_Dcm_00153

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType
BSW Parameter	BSW Type
UINT16	EcucEnumerationLiteralDef
BSW Description	Type of the data is uint16.

Template Description	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement  baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16	full
Mapping Status	Mapping ID
valid	up_Dcm_00205

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType
BSW Parameter	BSW Type
UINT16_N	EcucEnumerationLiteralDef
BSW Description	
Type of the data is uint16 array.	
Template Description	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::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 ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	Mapping ID

valid	up_Dcm_00132
-------	--------------

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType	
BSW Parameter	BSW Type	
UINT32	EcucEnumerationLiteralDef	
BSW Description		
Type of the data is uint32.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
Mapping Rule		Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement  baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32		full
Mapping Status		Mapping ID
valid		

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType	
BSW Parameter	BSW Type	
UINT32_N	EcucEnumerationLiteralDef	
BSW Description		
Type of the data is uint32 array.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.		
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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		

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements,	
<b>Mapping Rule</b>	<b>Mapping Type</b>
baseTypeEncoding = NONE, UTF-32 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
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00133

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT8	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the data is uint8.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement  baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT8_N	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the data is uint8 array.	
<b>Template Description</b>	



<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements,	
<b>Mapping Rule</b>	<b>Mapping Type</b>
baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP 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.fixedSize (cf. TPS_DEXT_01001)	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00172

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort
<b>BSW Parameter</b>	<b>BSW Type</b>
USE_DATA_SENDER_RECEIVER	EcucEnumerationLiteralDef
<b>BSW Description</b>	
<b>Template Description</b>	
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.	
<b>M2 Parameter</b>	
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping	
<b>Mapping Rule</b>	<b>Mapping Type</b>
DiagnosticServiceDataMapping.diagnosticDataElement refers to a DiagnosticDataElement that in turn is aggregated in the PID definition where the PortPrototype referenced in the role mappedDataElement.contextPort refers to a DataInterface where attribute isService is set to false.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00256

<b>BSW Module</b>	<b>BSW Context</b>
-------------------	--------------------

Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort	
<b>BSW Parameter</b>		<b>BSW Type</b>
USE_DATA_SENDER_RECEIVER_AS_SERVICE		EcucEnumerationLiteralDef
<b>BSW Description</b>		
<b>Template Description</b>		
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.		
<b>M2 Parameter</b>		
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping		
<b>Mapping Rule</b>		<b>Mapping Type</b>
DiagnosticServiceDataMapping.diagnosticDataElement refers to a Diagnostic DataElement that in turn is aggregated in the PID definition where the PortPrototype referenced in the role mappedDataElement.contextPort refers to a Data Interface where attribute isService is set to true.		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00257

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort	
<b>BSW Parameter</b>		<b>BSW Type</b>
USE_DATA_SYNCH_CLIENT_SERVER		EcucEnumerationLiteralDef
<b>BSW Description</b>		
<b>Template Description</b>		
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.		
<b>M2 Parameter</b>		
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping		
<b>Mapping Rule</b>		<b>Mapping Type</b>
DiagnosticServiceSwMapping.diagnosticDataElement refers to a Diagnostic DataElement that in turn is aggregated in the PID definition. DiagnosticServiceSwMapping.mappedSwcServiceDependency shall exist.		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00255

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort	
<b>BSW Parameter</b>		<b>BSW Type</b>
USE_DATA_SYNCH_FNC		EcucEnumerationLiteralDef
<b>BSW Description</b>		
<b>Template Description</b>		
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.		

<b>M2 Parameter</b>	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping	
<b>Mapping Rule</b>	<b>Mapping Type</b>
DiagnosticServiceSwMapping.diagnosticDataElement refers to a Diagnostic DataElement that in turn is aggregated in the PID definition and that also defines the role mappedBswServiceDependency.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00258

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidService01ExternalSRDataElementClass/DcmDataElement Instance
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDataElementInstanceRef	EcucInstanceReferenceDef
<b>BSW Description</b>	
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	
<b>Template Description</b>	
This represents the dataElement in the application software that is accessed for diagnostic purpose. This role is applicable on the classic platform.	
<b>M2 Parameter</b>	
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping.mappedDataElement	
<b>Mapping Rule</b>	<b>Mapping Type</b>
DiagnosticServiceDataMapping maps to a primitive data.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00100

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidService01ExternalSRDataElementClass/DcmSubElementInDataElementInstance
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmSubElementInDataElementInstanceRef	EcucInstanceReferenceDef
<b>BSW Description</b>	
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.	
<b>Template Description</b>	
This represents the dataElement in the application software that is accessed for diagnostic purpose. This role is applicable on the classic platform.	
<b>M2 Parameter</b>	
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping.mappedDataElement	
<b>Mapping Rule</b>	<b>Mapping Type</b>

DiagnosticServiceDataMapping maps to a primitive element within a composite data, where the AutosarDataPrototype is typed with a ApplicationComposite DataType.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00101

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidService01ExternalSRDataElementClass/DcmSubElementInImplDataElementInstance	
BSW Parameter		BSW Type
DcmSubElementInImplDataElementInstanceRef		EcucInstanceReferenceDef
BSW Description		
Instance Reference to the primitive sub-element (at any level) of composite data in a port which shall be read. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ImplementationDataType of category STRUCTURE or ARRAY. Please note that in case of ARRAY the index attribute in the target reference has to be set to select a single array element.		
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		
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping.mappedDataElement		
Mapping Rule		Mapping Type
DiagnosticServiceDataMapping maps to a primitive element within a composite data, where the AutosarDataPrototype is typed with a ApplicationComposite DataType ImplementationDataType of category STRUCTURE or ARRAY.		full
Mapping Status		Mapping ID
valid		up_Dcm_00102

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService02	
BSW Parameter		BSW Type
DcmDspPidDataDemRef		EcucReferenceDef
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		
DiagnosticExtract::Dcm::ObdService::Mode_0x02_RequestPowertrainFreezeFrameData::DiagnosticPowertrainFreezeFrame.pid		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid	
BSW Parameter	BSW Type	
DcmDspPidIdentifier	EcucIntegerParamDef	
BSW Description		
1 byte Identifier of the PID		
Within each DcmConfigSet all DcmDspPidIdentifier values shall be unique.		
Template Description		
<b>ObdPidServiceNeeds.parameterId:</b> Standardized parameter identifier (PID) according to the OBD standard specified in attribute "standard".		
<b>DiagnosticParameterIdentifier.id:</b> This is the numerical identifier used to identify the DiagnosticParameterIdentifier in the scope of diagnostic workflow (see SAE J1979-DA).		
M2 Parameter		
CommonStructure::ServiceNeeds::ObdPidServiceNeeds.parameterId, DiagnosticExtract::CommonDiagnostics::DiagnosticParameterIdentifier.id		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00028

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidService	
BSW Parameter	BSW Type	
DCM_SERVICE_01	EcucEnumerationLiteralDef	
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		
DiagnosticExtract::Dcm::ObdService::Mode_0x01_RequestCurrentPowertrainDiagnosticData:: DiagnosticRequestCurrentPowertrainData.pid		
Mapping Rule		Mapping Type
Applicable if the DiagnosticParameterIdentifier is only referenced by a DiagnosticRequestCurrentPowertrainData		full
Mapping Status		Mapping ID
valid		

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidService	
BSW Parameter	BSW Type	
DCM_SERVICE_01_02	EcucEnumerationLiteralDef	
BSW Description		
A PID is used with service \$01 and \$02. Allowed with a PID configuration containing data elements on byte basis.		
Template Description		

<b>DiagnosticRequestCurrentPowertrainData.pid:</b> This represents the PID associated with this instance of the OBD mode 0x01 service.	
<b>DiagnosticPowertrainFreezeFrame.pid:</b> This represents the PID associated with this instance of the OBD mode 0x02 service.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::ObdService::Mode_0x01_RequestCurrentPowertrainDiagnosticData::DiagnosticRequestCurrentPowertrainData.pid, DiagnosticExtract::Dcm::ObdService::Mode_0x02_RequestPowertrainFreezeFrameData::DiagnosticPowertrainFreezeFrame.pid	
<b>Mapping Rule</b>	<b>Mapping Type</b>
Applicable if the DiagnosticParameterIdentifier is referenced by both a DiagnosticRequestCurrentPowertrainData and a DiagnosticPowertrainFreezeFrame	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidService
<b>BSW Parameter</b>	<b>BSW Type</b>
DCM_SERVICE_02	EcucEnumerationLiteralDef
<b>BSW Description</b>	
A PID is used with service \$02 only. Allowed with a PID configuration containing data elements on byte basis.	
<b>Template Description</b>	
This represents the PID associated with this instance of the OBD mode 0x02 service.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::ObdService::Mode_0x02_RequestPowertrainFreezeFrameData::DiagnosticPowertrainFreezeFrame.pid	
<b>Mapping Rule</b>	<b>Mapping Type</b>
Applicable if the DiagnosticParameterIdentifier is only referenced by a DiagnosticPowertrainFreezeFrame	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspPidSize	EcucIntegerParamDef
<b>BSW Description</b>	
Length of a PID in byte(s).	
<b>Template Description</b>	
The size of the entire PID can be greater than the sum of the data elements because padding might be applied. Unit: byte.	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameterIdentifier.pidSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
-------------------	--------------------

Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspPidSupportInfo		EcucParamConfContainerDef
<b>BSW Description</b>		
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).		
<b>Template Description</b>		
This represents the supported information associated with the DiagnosticParameterIdentifier.		
<b>M2 Parameter</b>		
DiagnosticExtract::CommonDiagnostics::DiagnosticParameterIdentifier.supportInfoByte		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidSupportInfo	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspPidSupportInfoLen		EcucIntegerParamDef
<b>BSW Description</b>		
Length of the support information in bytes.		
<b>Template Description</b>		
This represents the size of the supportInfo within the PID. Unit: byte.		
<b>M2 Parameter</b>		
DiagnosticExtract::CommonDiagnostics::DiagnosticSupportInfoByte.size		
<b>Mapping Rule</b>		<b>Mapping Type</b>
		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidSupportInfo	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspPidSupportInfoPos		EcucIntegerParamDef
<b>BSW Description</b>		
Position of the support information in bytes.		
<b>Template Description</b>		
This represents the position of the supportInfo in the PID. Unit: byte.		
<b>M2 Parameter</b>		
DiagnosticExtract::CommonDiagnostics::DiagnosticSupportInfoByte.position		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRequestControl		EcucParamConfContainerDef
<b>BSW Description</b>		

This container contains the configuration (parameters) of the "Request control of on-board system, test or component" service (Service \$08).  
The DCM will request the control using an R-Port requiring a PortInterface RequestControlServices\_{Tid}.  
The R-Port is named RequestControlServices\_{Tid}  
where {Tid} is the name of the container DcmDspRequestControl.

<b>Template Description</b>	
This meta-class represents the ability to model an instance of the OBD mode 0x08 service.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::ObdService::Mode_0x08_RequestControlOfOnBoardDevice::DiagnosticRequestControlOfOnBoardDevice	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRequestControl
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspRequestControlInBufferSize	EcucIntegerParamDef
<b>BSW Description</b>	
Number of bytes to be provided in the input buffer of the interface RequestControlServices_{Tid} for OBD Service \$08	
<b>Template Description</b>	
This represents the specified data size for the request message. Unit: byte.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::ObdService::Mode_0x08_RequestControlOfOnBoardDevice::DiagnosticTestRoutineIdentifier.requestDataSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRequestControl
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspRequestControlOutBufferSize	EcucIntegerParamDef
<b>BSW Description</b>	
Number of bytes to be provided in the output buffer of the interface RequestControlServices_{Tid} for OBD Service \$08	
<b>Template Description</b>	
This represents the specified data size for the response message. Unit:byte.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::ObdService::Mode_0x08_RequestControlOfOnBoardDevice::DiagnosticTestRoutineIdentifier.responseDataSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
-------------------	--------------------



Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRequestControl	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRequestControlTestId		EcucIntegerParamDef
<b>BSW Description</b>		
Test Id for Service \$08		
<b>Template Description</b>		
<b>ObdControlServiceNeeds.testId:</b> Test Identifier (TID) according to ISO 15031-5.		
<b>DiagnosticTestRoutineIdentifier.id:</b> This represents the numerical id of the DiagnosticTestIdentifier (see SAE J1979-DA).		
<b>M2 Parameter</b>		
CommonStructure::ServiceNeeds::ObdControlServiceNeeds.testId, DiagnosticExtract::Dcm::ObdService::Mode_0x08_RequestControlOfOnBoardDevice::Diagnostic TestRoutineIdentifier.id		
<b>Mapping Rule</b>		<b>Mapping Type</b>
The value shall be taken from DiagnosticRequestControlOfOnBoardDevice.test Id.id if available.		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00030

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRequestFileTransfer		EcucParamConfContainerDef
<b>BSW Description</b>		
This container contains the configuration for RequestFileTransfer. This container only exists if RequestFileTransfer is configured.		
<b>Template Description</b>		
This diagnostic service instance implements the UDS service 0x38.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::RequestFileTransfer::DiagnosticRequestFileTransfer		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00135

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEvent/DcmDspRoe EventProperties/DcmDspRoeOnChangeOfDataIdentifier	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRoeDidRef		EcucReferenceDef
<b>BSW Description</b>		
Reference to a Did which is watched.		
<b>Template Description</b>		
This represents the corresponding DiagnosticDataIdentifier.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticDataChangeTrig- ger.dataIdentifier		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>

valid	up_Dcm_00136
-------	--------------

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEvent/DcmDspRoeEventProperties/DcmDspRoeOnDTCStatusChange	
BSW Parameter	BSW Type	
DcmDspRoeDTCStatusMask	EcucIntegerParamDef	
BSW Description	Value of the relevant DTCStatusMask	
Template Description	This attribute represents the ability to define a status mask for the triggering of an ROE response on the change of a DTC.	
M2 Parameter	DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticDtcChangeTrigger.dtcStatusMask	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00295	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEvent	
BSW Parameter	BSW Type	
DcmDspRoeInitialEventStatus	EcucEnumerationParamDef	
BSW Description	Initial Roe status of this RoeEvent	
Template Description	This represents the initial status of the enclosing DiagnosticResponseOnEventTrigger.	
M2 Parameter	DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticResponseOnEventTrigger.initialEventStatus	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00137	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEvent/DcmDspRoeInitialEventStatus	
BSW Parameter	BSW Type	
DCM_ROE_CLEARED	EcucEnumerationLiteralDef	
BSW Description		
Template Description	This means that the ResponseOnEvent is initially cleared.	
M2 Parameter	DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticInitialEventStatusEnum.returnOnEventCleared	
Mapping Rule	Mapping Type	
1:1 mapping	full	

Mapping Status	Mapping ID
valid	up_Dcm_00138

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEvent/DcmDspRoeInitialEventStatus	
BSW Parameter	BSW Type	
DCM_ROE_STOPPED	EcucEnumerationLiteralDef	
BSW Description		
Template Description		
This means that the ResponseOnEvent is initially stopped.		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticInitialEventStatus Enum.returnOnEventStopped		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00139	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe	
BSW Parameter	BSW Type	
DcmDspRoeEventWindowTime	EcucParamConfContainerDef	
BSW Description		
This container configures the available EventWindowTime in this Ecu.		
This container contains a sub-set of EventWindowTimes supported by the Dcm, to limit the Ecu resources.		
Template Description		
This attribute clarifies the validity of the eventWindow		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticEventWindow.eventWindowTime		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00140	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEventWindowTime/DcmDspRoeEventWindowTime	
BSW Parameter	BSW Type	
DCM_ROE_EVENT_WINDOW_CURRENT_AND_FOLLOWING_CYCLE	EcucEnumerationLiteralDef	
BSW Description		
Template Description		
This means that the window extends to this and the following cycle.		
M2 Parameter		

DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticEventWindowTime Enum.eventWindowCurrentAndFollowingCycle	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00141

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEventWindowTime/ DcmDspRoeEventWindowTime	
<b>BSW Parameter</b>		<b>BSW Type</b>
DCM_ROE_EVENT_WINDOW_CURRENT_CYCLE		EcucEnumerationLiteralDef
<b>BSW Description</b>		
<b>Template Description</b>		
This means that the window is limited to the current cycle.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticEventWindowTime Enum.eventWindowCurrentCycle		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00142

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEventWindowTime/ DcmDspRoeEventWindowTime	
<b>BSW Parameter</b>		<b>BSW Type</b>
DCM_ROE_EVENT_WINDOW_INFINITE		EcucEnumerationLiteralDef
<b>BSW Description</b>		
<b>Template Description</b>		
This means that the window extends without a border.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticEventWindowTime Enum.eventWindowInfinite		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00143

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRoeInterMessageTime		EcucFloatParamDef
<b>BSW Description</b>		
Provide the minimum time in seconds between two transmissions of ROE event. It is used for the delay between two different consecutive Roe transmissions.		
<b>Template Description</b>		
Provide the minimum time in seconds between two consecutive transmissions of an ROE event.		

<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticResponseOnEvent Class.interMessageTime	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00145

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspRoutine	EcucParamConfContainerDef
<b>BSW Description</b>	
This container contains the configuration (parameters) for Routines	
<b>Template Description</b>	
This meta-class represents the ability to define a diagnostic routine.	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticRoutine	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00181

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspRequestRoutineResults	EcucParamConfContainerDef
<b>BSW Description</b>	
Provides the configuration of RequestResult subservice for RoutineControl service. Existence indicates that the RequestRoutineResults in the RoutineControl is supported.	
<b>Template Description</b>	
<b>DiagnosticRoutine.requestResult:</b> This represents the ability to request the result of a running routine.	
<b>DiagnosticRoutineNeeds.diagRoutineType:</b> This denotes the type of diagnostic routine which is implemented by the referenced server port.	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticRoutine.requestResult, CommonStructure::ServiceNeeds::DiagnosticRoutineNeeds.diagRoutineType	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping for DiagnosticRoutine.requestResult  OR  DiagnosticRoutineNeeds.diagRoutineType == asynchronous	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00026

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineRe- sults

BSW Parameter		BSW Type
DcmDspRequestRoutineResultsFnc		EcucFunctionNameDef
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		
BswModuleTemplate::BswBehavior::BswServiceDependency		
Mapping Rule		Mapping Type
It could be possible to get the FNC name via BswServiceDependency		full
Mapping Status		Mapping ID
valid		up_Dcm_00147

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineRe- sults	
BSW Parameter		BSW Type
DcmDspRequestRoutineResultsIn		EcucParamConfContainerDef
BSW Description		
Provide description of input parameter of RequestResult subservice for RoutineControl service.		
Template Description		
This represents the request parameters.		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticRequestRoutineResults.request		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00296

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineRe- sults/DcmDspRequestRoutineResultsIn	
BSW Parameter		BSW Type
DcmDspRequestRoutineResultsInSignal		EcucParamConfContainerDef
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		
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.dataElement		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00149

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType	
BSW Parameter		BSW Type
DcmDspTextTableMapping		EcucParamConfContainerDef
BSW Description		
<p>The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists).</p> <p>In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.</p>		
Template Description		
<p>This meta-class represents the ability to express the relationship between a physical value and the mathematical representation.</p> <p>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.</p>		
M2 Parameter		
AsamHdo::ComputationMethod::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		Mapping ID
valid		up_Dcm_00097

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping	
BSW Parameter		BSW Type
DcmDspDiagnosisRepresentationDataValue		EcucIntegerParamDef
BSW Description		
The data value in the diagnosis representation.		
Template Description		
This represents a textual constant in the computation method.		
M2 Parameter		
AsamHdo::ComputationMethod::CompuConstTextContent.vt		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00098

BSW Module	BSW Context
------------	-------------

Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspInternalDataValue		EcucIntegerParamDef
<b>BSW Description</b>		
The ECU internal data value.		
<b>Template Description</b>		
<b>CompuScale.lowerLimit:</b> This specifies the lower limit of the scale.		
<b>CompuScale.upperLimit:</b> This specifies the upper limit of a of the scale.		
<b>M2 Parameter</b>		
AsamHdo::ComputationMethod::CompuScale.lowerLimit, AsamHdo::ComputationMethod::CompuScale.upperLimit		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00099

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRoutineParameterSize		EcucIntegerParamDef
<b>BSW Description</b>		
Provide the size of a RoutineControl parameter in bytes		
<b>Template Description</b>		
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.		
<b>M2 Parameter</b>		
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements		
<b>Mapping Rule</b>		<b>Mapping Type</b>
Only in case of variable length required (according to constr_6008). Calculation: $DcmDspRoutineSignalLength = \maxNumberOfElements * 8$		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00189

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRoutineSignalEndianness		EcucEnumerationParamDef
<b>BSW Description</b>		
Defines the endianness of the data belonging to a Routine In Signal for RequestResult subfunction.		
<b>Template Description</b>		
This attribute specifies the byte order of the base type.		
<b>M2 Parameter</b>		



AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder	
<b>Mapping Rule</b>	<b>Mapping Type</b>
baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00185

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRoutineSignalPos		EcucIntegerParamDef
<b>BSW Description</b>		
Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.		
<b>Template Description</b>		
This represents the bitOffset of the DiagnosticParameter		
<b>M2 Parameter</b>		
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.bitOffset		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00193

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspRoutineSignalType	
<b>BSW Parameter</b>		<b>BSW Type</b>
BOOLEAN		EcucEnumerationLiteralDef
<b>BSW Description</b>		
Type of the signal is boolean.		
<b>Template Description</b>		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>M2 Parameter</b>		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
<b>Mapping Rule</b>		<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response		full
baseTypeEncoding = BOOLEAN baseTypeSize = 1		
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00199

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspRoutineSignalType	
BSW Parameter		BSW Type
SINT16		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is sint16.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
Mapping Rule		Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response  baseTypeEncoding = 2C baseTypeSize = 16		full
Mapping Status		Mapping ID
valid		up_Dcm_00156

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspRoutineSignalType	
BSW Parameter		BSW Type
SINT16_N		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is sint16 array.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.		
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.		
M2 Parameter		

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
<b>Mapping Rule</b>	<b>Mapping Type</b>
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 ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00156

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
SINT32	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is sint32.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response  baseTypeEncoding = 2C baseTypeSize = 32	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00163

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
SINT32_N	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is sint32 array.	

Template Description	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::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	Mapping ID
valid	up_Dcm_00163

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspRoutineSignalType
BSW Parameter	BSW Type
SINT8	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint8.	
Template Description	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	Mapping Type

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response	full
baseTypeEncoding = 2C baseTypeSize = 8	
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00201

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspRoutineSignalType
BSW Parameter	BSW Type
SINT8_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint8 array.	
Template Description	
<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p> <p><b>DiagnosticDataElement.maxNumberOfElements:</b> 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.</p> <p><b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.</p>	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response	full
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.fixedSize (cf. TPS_DEXT_01001)	
Mapping Status	Mapping ID
valid	up_Dcm_00201

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspRoutineSignalType

BSW Parameter		BSW Type
UINT16		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is uint16.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
Mapping Rule		Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response  baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16		full
Mapping Status		Mapping ID
valid		up_Dcm_00207

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspRoutineSignalType	
BSW Parameter		BSW Type
UINT16_N		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is uint16 array.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.		
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements		
Mapping Rule		Mapping Type

<p>referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response</p> <p>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 ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)</p>	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00207

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspRoutineSignalType	
BSW Parameter		BSW Type
UINT32		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is uint32.		
Template Description		
<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p>		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
Mapping Rule		Mapping Type
<p>referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response</p> <p>baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32</p>		full
Mapping Status		Mapping ID
valid		up_Dcm_00210

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspRoutineSignalType	
BSW Parameter		BSW Type
UINT32_N		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is uint32 array.		
Template Description		

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response  baseTypeEncoding = NONE, UTF-32, UTF-32 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
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00210

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT8	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is uint8.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>



referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response	full
baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8	
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00167

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspRoutineSignalType
BSW Parameter	BSW Type
UINT8_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint8 array.	
Template Description	
<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p> <p><b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.</p> <p><b>DiagnosticDataElement.maxNumberOfElements:</b> 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.</p>	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response	full
baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP 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.fixedSize (cf. TPS_DEXT_01001)	
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00167

BSW Module	BSW Context
------------	-------------

Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal/DcmDspRoutineSignalType	
<b>BSW Parameter</b>		<b>BSW Type</b>
VARIABLE_LENGTH		EcucEnumerationLiteralDef
<b>BSW Description</b>		
Type of the signal is uint8[DcmDspRoutineParameterSize].		
This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.		
<b>Template Description</b>		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.		
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.		
<b>M2 Parameter</b>		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics		
<b>Mapping Rule</b>		<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response		full
baseTypeEncoding = NONE baseTypeSize = 8 arraySizeSemantics = variableSize maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01002) arraySizeSemantics exists and is set to ArraySizeSemanticsEnum.variableSize (cf. TPS_DEXT_01002)		
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00217

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults
<b>BSW Parameter</b>	
DcmDspRequestRoutineResultsOut	
<b>BSW Type</b>	
EcucParamConfContainerDef	
<b>BSW Description</b>	
Provide description of output parameter of RequestResult subservice for RoutineControl service.	
<b>Template Description</b>	
This represents the response parameters.	
<b>M2 Parameter</b>	

DiagnosticExtract::CommonDiagnostics::DiagnosticRequestRoutineResults.response	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00148

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRequestRoutineResultsOutSignal		EcucParamConfContainerDef
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
This represents the related dataElement of the DiagnosticParameter		
<b>M2 Parameter</b>		
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.dataElement		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00149

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspTextTableMapping		EcucParamConfContainerDef
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
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.		
<b>M2 Parameter</b>		
AsamHdo::ComputationMethod::CompuMethod		
<b>Mapping Rule</b>		<b>Mapping Type</b>
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.		full

Mapping Status	Mapping ID
valid	up_Dcm_00097

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping	
BSW Parameter		BSW Type
DcmDspDiagnosisRepresentationDataValue		EcucIntegerParamDef
BSW Description		
The data value in the diagnosis representation.		
Template Description		
This represents a textual constant in the computation method.		
M2 Parameter		
AsamHdo::ComputationMethod::CompuConstTextContent.vt		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00098

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping	
BSW Parameter		BSW Type
DcmDspInternalDataValue		EcucIntegerParamDef
BSW Description		
The ECU internal data value.		
Template Description		
<b>CompuScale.lowerLimit:</b> This specifies the lower limit of the scale.		
<b>CompuScale.upperLimit:</b> This specifies the upper limit of a of the scale.		
M2 Parameter		
AsamHdo::ComputationMethod::CompuScale.lowerLimit, AsamHdo::ComputationMethod::CompuScale.upperLimit		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00099

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal	
BSW Parameter		BSW Type
DcmDspRoutineParameterSize		EcucIntegerParamDef
BSW Description		

Provide the size of a RoutineControl parameter in bytes	
<b>Template Description</b>	
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.	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00189

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOut Signal	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRoutineSignalEndianness		EcucEnumerationParamDef
<b>BSW Description</b>		
Defines the endianness of the data belonging to a Routine Out Signal for RequestResult subfunction.		
<b>Template Description</b>		
This attribute specifies the byte order of the base type.		
<b>M2 Parameter</b>		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder		
<b>Mapping Rule</b>		<b>Mapping Type</b>
baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResult.response		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00185

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOut Signal	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRoutineSignalPos		EcucIntegerParamDef
<b>BSW Description</b>		
Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.		
<b>Template Description</b>		
This represents the bitOffset of the DiagnosticParameter		
<b>M2 Parameter</b>		
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.bitOffset		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00193

<b>BSW Module</b>	<b>BSW Context</b>
-------------------	--------------------

Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType	
<b>BSW Parameter</b>		<b>BSW Type</b>
BOOLEAN		EcucEnumerationLiteralDef
<b>BSW Description</b>		
Type of the signal is boolean.		
<b>Template Description</b>		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>M2 Parameter</b>		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
<b>Mapping Rule</b>		<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response  baseTypeEncoding = BOOLEAN baseTypeSize = 1		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00199

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType	
<b>BSW Parameter</b>		<b>BSW Type</b>
SINT16		EcucEnumerationLiteralDef
<b>BSW Description</b>		
Type of the signal is sint16.		
<b>Template Description</b>		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>M2 Parameter</b>		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
<b>Mapping Rule</b>		<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response  baseTypeEncoding = 2C baseTypeSize = 16		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00156

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType	
BSW Parameter		BSW Type
SINT16_N		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is sint16 array.		
Template Description		
<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p> <p><b>DiagnosticDataElement.maxNumberOfElements:</b> 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.</p> <p><b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.</p>		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics		
Mapping Rule		Mapping Type
<p>referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response</p> <p>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 ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)</p>		full
Mapping Status		Mapping ID
valid		up_Dcm_00156

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType	
BSW Parameter		BSW Type
SINT32		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is sint32.		
Template Description		

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response  baseTypeEncoding = 2C baseTypeSize = 32	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00163

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
SINT32_N	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is sint32 array.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
<b>Mapping Rule</b>	<b>Mapping Type</b>



<p>referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response</p> <p>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)</p>	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00163

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType	
BSW Parameter		BSW Type
SINT8		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is sint8.		
Template Description		
<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p>		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
Mapping Rule		Mapping Type
<p>referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response</p> <p>baseTypeEncoding = 2C baseTypeSize = 8</p>		full
Mapping Status		Mapping ID
valid		up_Dcm_00201

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType	
BSW Parameter		BSW Type
SINT8_N		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is sint8 array.		
Template Description		

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
<b>Mapping Rule</b>	<b>Mapping Type</b>
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 ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00201

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT16	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is uint16.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response	full
baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16	
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00207

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	BSW Type
UINT16_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint16 array.	
Template Description	
<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p> <p><b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.</p> <p><b>DiagnosticDataElement.maxNumberOfElements:</b> 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.</p>	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response	full
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 ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	
<b>Mapping Status</b>	
valid	<b>Mapping ID</b> up_Dcm_00207

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter		BSW Type
UINT32		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is uint32.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
Mapping Rule		Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response  baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32		full
Mapping Status		Mapping ID
valid		up_Dcm_00210

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType	
BSW Parameter		BSW Type
UINT32_N		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is uint32 array.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.		
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements		
Mapping Rule		Mapping Type

<p>referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response</p> <p>baseTypeEncoding = NONE, UTF-32, UTF-32 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)</p>	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00210

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
UINT8	EcucEnumerationLiteralDef	
BSW Description	Type of the signal is uint8.	
Template Description	<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p>	
M2 Parameter	AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	Mapping Type	
<p>referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response</p> <p>baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8</p>	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00167	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
UINT8_N	EcucEnumerationLiteralDef	
BSW Description	Type of the signal is uint8 array.	
Template Description		

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response  baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP 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.fixedSize (cf. TPS_DEXT_01001)	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00167

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
VARIABLE_LENGTH	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is uint8[DcmDspRoutineParameterSize].  This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.	
<b>Template Description</b>	

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response  baseTypeEncoding = NONE baseTypeSize = 8 arraySizeSemantics = variableSize 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
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00217

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspRoutineIdentifier	EcucIntegerParamDef
<b>BSW Description</b>	
2 bytes Identifier of the RID  Within each DcmConfigSet all DcmDspRoutineIdentifier values shall be unique.	
<b>Template Description</b>	
<b>DiagnosticRoutine.id:</b> This is the numerical identifier used to identify the DiagnosticRoutine in the scope of diagnostic workflow	
<b>DiagnosticRoutineNeeds.ridNumber:</b> This represents a routine identifier for the diagnostic routine. This allows to predefine the RID number if the a function developer has received a particular requirement from the OEM or from a standardization body.	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticRoutine.id, CommonStructure::ServiceNeeds::DiagnosticRoutineNeeds.ridNumber	
<b>Mapping Rule</b>	<b>Mapping Type</b>

1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00003

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine	
BSW Parameter	BSW Type	
DcmDspRoutineUsePort	EcucBooleanParamDef	
BSW Description		
<p>If this parameter is set to true, the DCM uses a port requiring a PortInterface RoutineServices_{RoutineName}.</p> <p>The R-Port is named RoutineServices_{RoutineName} where {RoutineName} is the name of the container DcmDspRoutine. In that case, the configuration must not provide function names in DcmDspStartRoutineFnc, DcmDspStopRoutineFnc or DcmDspRequestResultsRoutineFnc. If this is false, the DCM expects to find the names of the functions to be used in DcmDspStartRoutineFnc, DcmDspStopRoutineFnc or DcmDspRequestResultsRoutineFnc.</p>		
Template Description		
<p>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.</p>		
M2 Parameter		
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping		
Mapping Rule	Mapping Type	
TRUE: DiagnosticServiceSwMapping is having a SwcServiceDependency FALSE: DiagnosticServiceSwMapping is having a BswServiceDependency	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00174	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine	
BSW Parameter	BSW Type	
DcmDspStartRoutine	EcucParamConfContainerDef	
BSW Description		
<p>Provides the configuration of Start subservice for RoutineControl service.</p>		
Template Description		
<p><b>DiagnosticRoutine.start:</b> This represents the ability to start a routine</p> <p><b>DiagnosticRoutineNeeds:</b> 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.</p>		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticRoutine.start, CommonStructure::ServiceNeeds::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	Mapping ID	
valid	up_Dcm_00024	



BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine	
BSW Parameter	BSW Type	
DcmDspStartRoutineFnc	EcucFunctionNameDef	
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	DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00175	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine	
BSW Parameter	BSW Type	
DcmDspStartRoutineIn	EcucParamConfContainerDef	
BSW Description	Provide description of input parameter of Start subservice for RoutineControl service	
Template Description	This represents the request parameters.	
M2 Parameter	DiagnosticExtract::CommonDiagnostics::DiagnosticStartRoutine.request	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00176	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn	
BSW Parameter	BSW Type	
DcmDspStartRoutineInSignal	EcucParamConfContainerDef	
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	DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.dataElement	
Mapping Rule	Mapping Type	
1:1 mapping	full	

Mapping Status	Mapping ID
valid	up_Dcm_00177

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType	
BSW Parameter		BSW Type
DcmDspTextTableMapping		EcucParamConfContainerDef
BSW Description		
<p>The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists).</p> <p>In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.</p>		
Template Description		
<p>This meta-class represents the ability to express the relationship between a physical value and the mathematical representation.</p> <p>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.</p>		
M2 Parameter		
AsamHdo::ComputationMethod::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		Mapping ID
valid		up_Dcm_00097

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping	
BSW Parameter		BSW Type
DcmDspDiagnosisRepresentationDataValue		EcucIntegerParamDef
BSW Description		
The data value in the diagnosis representation.		
Template Description		
This represents a textual constant in the computation method.		
M2 Parameter		
AsamHdo::ComputationMethod::CompuConstTextContent.vt		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00098

BSW Module	BSW Context
------------	-------------

Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspInternalDataValue		EcucIntegerParamDef
<b>BSW Description</b>		
The ECU internal data value.		
<b>Template Description</b>		
<b>CompuScale.lowerLimit:</b> This specifies the lower limit of the scale.		
<b>CompuScale.upperLimit:</b> This specifies the upper limit of a of the scale.		
<b>M2 Parameter</b>		
AsamHdo::ComputationMethod::CompuScale.lowerLimit, AsamHdo::ComputationMethod::CompuScale.upperLimit		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00099

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRoutineParameterSize		EcucIntegerParamDef
<b>BSW Description</b>		
Provide the size of a RoutineControl parameter in bytes		
<b>Template Description</b>		
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.		
<b>M2 Parameter</b>		
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements		
<b>Mapping Rule</b>		<b>Mapping Type</b>
"Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8.		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00186

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRoutineSignalEndianness		EcucEnumerationParamDef
<b>BSW Description</b>		
Defines the endianness of the data belonging to a Routine In Signal for Start subfunction.		
<b>Template Description</b>		
This attribute specifies the byte order of the base type.		
<b>M2 Parameter</b>		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder		
<b>Mapping Rule</b>		<b>Mapping Type</b>

baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00182

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal	
BSW Parameter		BSW Type
DcmDspRoutineSignalPos		EcucIntegerParamDef
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		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.bitOffset		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00190

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType	
BSW Parameter		BSW Type
BOOLEAN		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is boolean.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
Mapping Rule		Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request  baseTypeEncoding = BOOLEAN baseTypeSize = 1		full
Mapping Status		Mapping ID
valid		up_Dcm_00196

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType	

BSW Parameter		BSW Type
SINT16		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is sint16.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
Mapping Rule		Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request  baseTypeEncoding = 2C baseTypeSize = 16		full
Mapping Status		Mapping ID
valid		up_Dcm_00155

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType	
BSW Parameter		BSW Type
SINT16_N		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is sint16 array.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.		
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics		
Mapping Rule		Mapping Type

<p>referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request</p> <p>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 ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)</p>	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00155

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
SINT32	EcucEnumerationLiteralDef	
BSW Description	Type of the signal is sint32.	
Template Description	<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p>	
M2 Parameter	AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	Mapping Type	
<p>referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request</p> <p>baseTypeEncoding = 2C baseTypeSize = 32</p>	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00164	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
SINT32_N	EcucEnumerationLiteralDef	
BSW Description	Type of the signal is sint32 array.	
Template Description		

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request  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
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00164

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
SINT8	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is sint8.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request  baseTypeEncoding = 2C baseTypeSize = 8	full

Mapping Status	Mapping ID
valid	up_Dcm_00200

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	BSW Type
SINT8_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint8 array.	
Template Description	
<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p> <p><b>DiagnosticDataElement.maxNumberOfElements:</b> 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.</p> <p><b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.</p>	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	Mapping Type
<p>referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request</p> <p>baseTypeEncoding = 2C baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either doesnot exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)</p>	full
Mapping Status	Mapping ID
valid	up_Dcm_00200

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	BSW Type
UINT16	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint16.	
Template Description	



<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request  baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00204

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT16_N	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is uint16 array.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request  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 ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full

Mapping Status	Mapping ID
valid	up_Dcm_00204

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
UINT32	EcucEnumerationLiteralDef	
BSW Description		
Type of the signal is uint32.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
Mapping Rule	Mapping Type	
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request	full	
baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32		
Mapping Status	Mapping ID	
valid	up_Dcm_00213	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
UINT32_N	EcucEnumerationLiteralDef	
BSW Description		
Type of the signal is uint32 array.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.		
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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		

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request  baseTypeEncoding = NONE, UTF-32  maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEX T_01001) arraySizeSemantics either does not exist or exists and is set to ArraySize SemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00213

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDsp StartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT8	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is uint8.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request  baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-U P baseTypeSize = 8	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00169

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDsp StartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT8_N	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is uint8 array.	
<b>Template Description</b>	

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request  baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP 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.fixedSize (cf. TPS_DEXT_01001)	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00169

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
VARIABLE_LENGTH	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is uint8[DcmDspRoutineParameterSize].  This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.	
<b>Template Description</b>	

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request  baseTypeEncoding = NONE baseTypeSize = 8 arraySizeSemantics = variableSize 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
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00219

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspStartRoutineOut	EcucParamConfContainerDef
<b>BSW Description</b>	
Provide description of output parameter of Start subservice for RoutineControl service.	
<b>Template Description</b>	
This represents the response parameters.	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticStartRoutine.response	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00243

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspStartRoutineOutSignal	EcucParamConfContainerDef

<b>BSW Description</b>	
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.	
<b>Template Description</b>	
This represents the related dataElement of the DiagnosticParameter	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.dataElement	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00242

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType	
<b>BSW Parameter</b>	<b>BSW Type</b>	
DcmDspTextTableMapping	EcucParamConfContainerDef	
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
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.		
<b>M2 Parameter</b>		
AsamHdo::ComputationMethod::CompuMethod		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid	up_Dcm_00097	

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping	
<b>BSW Parameter</b>	<b>BSW Type</b>	
DcmDspDiagnosisRepresentationDataValue	EcucIntegerParamDef	
<b>BSW Description</b>		
The data value in the diagnosis representation.		

Template Description	
This represents a textual constant in the computation method.	
M2 Parameter	
AsamHdo::ComputationMethod::CompuConstTextContent.vt	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	Mapping ID
valid	up_Dcm_00098

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping	
BSW Parameter		BSW Type
DcmDspInternalDataValue		EcucIntegerParamDef
BSW Description		
The ECU internal data value.		
Template Description		
<b>CompuScale.lowerLimit:</b> This specifies the lower limit of the scale.		
<b>CompuScale.upperLimit:</b> This specifies the upper limit of a of the scale.		
M2 Parameter		
AsamHdo::ComputationMethod::CompuScale.lowerLimit, AsamHdo::ComputationMethod::CompuScale.upperLimit		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00099	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal	
BSW Parameter		BSW Type
DcmDspRoutineParameterSize		EcucIntegerParamDef
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		
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements		
Mapping Rule	Mapping Type	
Only in case of variable length required (according to constr_6008). Calculation: $DcmDspRoutineSignalLength = maxNumberOfElements * 8$ .	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00151	

BSW Module	BSW Context
------------	-------------

Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRoutineSignalEndianness		EcucEnumerationParamDef
<b>BSW Description</b>		
Defines the endianness of the data belonging to a Routine Out Signal for Start subfunction.		
<b>Template Description</b>		
This attribute specifies the byte order of the base type.		
<b>M2 Parameter</b>		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder		
<b>Mapping Rule</b>		<b>Mapping Type</b>
baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00150

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRoutineSignalPos		EcucIntegerParamDef
<b>BSW Description</b>		
Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.		
<b>Template Description</b>		
This represents the bitOffset of the DiagnosticParameter		
<b>M2 Parameter</b>		
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.bitOffset		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00152

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType	
<b>BSW Parameter</b>		<b>BSW Type</b>
BOOLEAN		EcucEnumerationLiteralDef
<b>BSW Description</b>		
Type of the signal is boolean.		
<b>Template Description</b>		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>M2 Parameter</b>		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
<b>Mapping Rule</b>		<b>Mapping Type</b>



referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response	full
baseTypeEncoding = BOOLEAN baseTypeSize = 1	
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00198

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
SINT16	EcucEnumerationLiteralDef	
BSW Description		
Type of the signal is sint16.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
Mapping Rule	Mapping Type	
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response	full	
baseTypeEncoding = 2C baseTypeSize = 16		
Mapping Status	Mapping ID	
valid	up_Dcm_00157	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
SINT16_N	EcucEnumerationLiteralDef	
BSW Description		
Type of the signal is sint16 array.		
Template Description		

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.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 ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00157

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
SINT32	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is sint32.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response  baseTypeEncoding = 2C baseTypeSize = 32	full

Mapping Status	Mapping ID
valid	up_Dcm_00161

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
SINT32_N	EcucEnumerationLiteralDef	
BSW Description		
Type of the signal is sint32 array.		
Template Description		
<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p> <p><b>DiagnosticDataElement.maxNumberOfElements:</b> 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.</p> <p><b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.</p>		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics		
Mapping Rule		Mapping Type
<p>referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response</p> <p>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)</p>		full
Mapping Status		Mapping ID
valid		up_Dcm_00161

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
SINT8	EcucEnumerationLiteralDef	
BSW Description		
Type of the signal is sint8.		
Template Description		

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response  baseTypeEncoding = 2C baseTypeSize = 8	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00202

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
SINT8_N	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is sint8 array.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.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 ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full

Mapping Status	Mapping ID
valid	up_Dcm_00202

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
UINT16	EcucEnumerationLiteralDef	
BSW Description		
Type of the signal is uint16.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
Mapping Rule	Mapping Type	
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response  baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00206	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
UINT16_N	EcucEnumerationLiteralDef	
BSW Description		
Type of the signal is uint16 array.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.		
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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		

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response  baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemanticseither does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00206

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT32	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is uint32.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response  baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00215

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT32_N	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is uint32 array.	
<b>Template Description</b>	

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response  baseTypeEncoding = NONE, UTF-32 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
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00215

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT8	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is uint8.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response	full
baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8	
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00168

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	BSW Type
UINT8_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint8 array.	
Template Description	
<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p> <p><b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.</p> <p><b>DiagnosticDataElement.maxNumberOfElements:</b> 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.</p>	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response	full
baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP 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.fixedSize (cf. TPS_DEXT_01001)	
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00168

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType



BSW Parameter	BSW Type
VARIABLE_LENGTH	EcucEnumerationLiteralDef
BSW Description	
<p>Type of the signal is uint8[DcmDspRoutineParameterSize].</p> <p>This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.</p>	
Template Description	
<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p> <p><b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.</p> <p><b>DiagnosticDataElement.maxNumberOfElements:</b> 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.</p>	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response  baseTypeEncoding = NONE baseTypeSize = 8 arraySizeSemantics = variableSize 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	Mapping ID
valid	up_Dcm_00216

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine
BSW Parameter	BSW Type
DcmDspStopRoutine	EcucParamConfContainerDef
BSW Description	
<p>Provides the configuration of Stop subservice for RoutineControl service. Existence indicates that the StopRoutine in the RoutineControl is supported.</p>	
Template Description	
<p><b>DiagnosticRoutine.stop:</b> This represents the ability to stop a running routine.</p> <p><b>DiagnosticRoutineNeeds.diagRoutineType:</b> This denotes the type of diagnostic routine which is implemented by the referenced server port.</p>	

<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticRoutine.stop, CommonStructure::ServiceNeeds::DiagnosticRoutineNeeds.diagRoutineType	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping for DiagnosticRoutine.stop  OR  DiagnosticRoutineNeeds.diagRoutineType == asynchronous	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00025

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspStopRoutineFnc	EcucFunctionNameDef
<b>BSW Description</b>	
Function name for request to application to stop a routine. (Routine_Stop-function)	
This parameter is related to the interface Xxx_Stop.	
<b>Template Description</b>	
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.	
<b>M2 Parameter</b>	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00221

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspStopRoutineIn	EcucParamConfContainerDef
<b>BSW Description</b>	
Provide description of input parameter of Stop subservice for RoutineControl service.	
<b>Template Description</b>	
This represents the request parameters.	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticStopRoutine.request	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00220

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn
<b>BSW Parameter</b>	<b>BSW Type</b>

DcmDspStopRoutineInSignal	EcucParamConfContainerDef
<b>BSW Description</b>	
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.	
<b>Template Description</b>	
This represents the related dataElement of the DiagnosticParameter	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.dataElement	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00222

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType	
<b>BSW Parameter</b>	<b>BSW Type</b>	
DcmDspTextTableMapping	EcucParamConfContainerDef	
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
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.		
<b>M2 Parameter</b>		
AsamHdo::ComputationMethod::CompuMethod		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid	up_Dcm_00097	

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping	
<b>BSW Parameter</b>	<b>BSW Type</b>	
DcmDspDiagnosisRepresentationDataValue	EcucIntegerParamDef	
<b>BSW Description</b>		

The data value in the diagnosis representation.	
<b>Template Description</b>	
This represents a textual constant in the computation method.	
<b>M2 Parameter</b>	
AsamHdo::ComputationMethod::CompuConstTextContent.vt	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00098

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspInternalDataValue		EcucIntegerParamDef
<b>BSW Description</b>		
The ECU internal data value.		
<b>Template Description</b>		
<b>CompuScale.lowerLimit:</b> This specifies the lower limit of the scale.		
<b>CompuScale.upperLimit:</b> This specifies the upper limit of a of the scale.		
<b>M2 Parameter</b>		
AsamHdo::ComputationMethod::CompuScale.lowerLimit, AsamHdo::ComputationMethod::CompuScale.upperLimit		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
1:1 mapping	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid	up_Dcm_00099	

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRoutineParameterSize		EcucIntegerParamDef
<b>BSW Description</b>		
Provide the size of a RoutineControl parameter in bytes		
<b>Template Description</b>		
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.		
<b>M2 Parameter</b>		
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
Only in case of variable length required (according to constr_6008). Calculation: $DcmDspRoutineSignalLength = \text{maxNumberOfElements} * 8$	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid	up_Dcm_00188	

<b>BSW Module</b>	<b>BSW Context</b>
-------------------	--------------------

Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRoutineSignalEndianness		EcucEnumerationParamDef
<b>BSW Description</b>		
Defines the endianness of the data belonging to a Routine In Signal for Stop subfunction.		
<b>Template Description</b>		
This attribute specifies the byte order of the base type.		
<b>M2 Parameter</b>		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder		
<b>Mapping Rule</b>		<b>Mapping Type</b>
baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00183

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspRoutineSignalPos		EcucIntegerParamDef
<b>BSW Description</b>		
Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.		
<b>Template Description</b>		
This represents the bitOffset of the DiagnosticParameter		
<b>M2 Parameter</b>		
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.bitOffset		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00192

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType	
<b>BSW Parameter</b>		<b>BSW Type</b>
BOOLEAN		EcucEnumerationLiteralDef
<b>BSW Description</b>		
Type of the signal is boolean.		
<b>Template Description</b>		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>M2 Parameter</b>		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
<b>Mapping Rule</b>		<b>Mapping Type</b>

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request	full
baseTypeEncoding = BOOLEAN baseTypeSize = 1	
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00194

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
SINT16	EcucEnumerationLiteralDef	
BSW Description		
Type of the signal is sint16.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
Mapping Rule	Mapping Type	
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request	full	
baseTypeEncoding = 2C baseTypeSize = 16		
Mapping Status	Mapping ID	
valid	up_Dcm_00158	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
SINT16_N	EcucEnumerationLiteralDef	
BSW Description		
Type of the signal is sint16 array.		
Template Description		

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request  baseTypeEncoding = 2C baseTypeSize = 16 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either doesnot exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00158

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
SINT32	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is sint32.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request  baseTypeEncoding = 2C baseTypeSize = 32	full

Mapping Status	Mapping ID
valid	up_Dcm_00165

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
SINT32_N	EcucEnumerationLiteralDef	
BSW Description	Type of the signal is sint32 array.	
Template Description	<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p> <p><b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.</p> <p><b>DiagnosticDataElement.maxNumberOfElements:</b> 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.</p>	
M2 Parameter	AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	Mapping Type	
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request  baseTypeEncoding = 2C baseTypeSize = 32 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either doesnot exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00165	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
SINT8	EcucEnumerationLiteralDef	
BSW Description	Type of the signal is sint8.	
Template Description		



<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request  baseTypeEncoding = 2C baseTypeSize = 8	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00247

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
SINT8_N	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is sint8 array.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request  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.fixedSize (cf. TPS_DEXT_01001)	full

Mapping Status	Mapping ID
valid	up_Dcm_00247

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
UINT16	EcucEnumerationLiteralDef	
BSW Description		
Type of the signal is uint16.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
Mapping Rule	Mapping Type	
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request  baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00208	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType	
BSW Parameter	BSW Type	
UINT16_N	EcucEnumerationLiteralDef	
BSW Description		
Type of the signal is uint16 array.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.		
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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		

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request  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 ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00208

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT32	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is uint32.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00214

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT32_N	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is uint32 array.	
<b>Template Description</b>	

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
baseTypeEncoding = NONE, UTF-32 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
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00214

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT8	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is uint8.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request  baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00171

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType	
BSW Parameter		BSW Type
UINT8_N		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is uint8 array.		
Template Description		
<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p> <p><b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.</p> <p><b>DiagnosticDataElement.maxNumberOfElements:</b> 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.</p>		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements		
Mapping Rule		Mapping Type
<p>referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request</p> <p>baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP 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.fixedSize (cf. TPS_DEXT_01001)</p>		full
Mapping Status		Mapping ID
valid		up_Dcm_00171

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType	
BSW Parameter		BSW Type
VARIABLE_LENGTH		EcucEnumerationLiteralDef
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		

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request  baseTypeEncoding = NONE baseTypeSize = 8 arraySizeSemantics = variableSize 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
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00218

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspStopRoutineOut		EcucParamConfContainerDef
<b>BSW Description</b>		
Provide description of output parameter of Stop subservice for RoutineControl service.		
<b>Template Description</b>		
This represents the response parameters.		
<b>M2 Parameter</b>		
DiagnosticExtract::CommonDiagnostics::DiagnosticStopRoutine.response		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00224

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspStopRoutineOutSignal		EcucParamConfContainerDef
<b>BSW Description</b>		

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.	
<b>Template Description</b>	
This represents the related dataElement of the DiagnosticParameter	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.dataElement	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00223

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspTextTableMapping	EcucParamConfContainerDef
<b>BSW Description</b>	
<p>The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists).</p> <p>In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.</p>	
<b>Template Description</b>	
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.	
<b>M2 Parameter</b>	
AsamHdo::ComputationMethod::CompuMethod	
<b>Mapping Rule</b>	<b>Mapping Type</b>
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00097

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspDiagnosisRepresentationDataValue	EcucIntegerParamDef
<b>BSW Description</b>	
The data value in the diagnosis representation.	
<b>Template Description</b>	

This represents a textual constant in the computation method.	
<b>M2 Parameter</b>	
AsamHdo::ComputationMethod::CompuConstTextContent.vt	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00098

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspInternalDataValue	EcucIntegerParamDef
<b>BSW Description</b>	
The ECU internal data value.	
<b>Template Description</b>	
<b>CompuScale.lowerLimit:</b> This specifies the lower limit of the scale.	
<b>CompuScale.upperLimit:</b> This specifies the upper limit of a of the scale.	
<b>M2 Parameter</b>	
AsamHdo::ComputationMethod::CompuScale.lowerLimit, AsamHdo::ComputationMethod::CompuScale.upperLimit	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00099

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspRoutineParameterSize	EcucIntegerParamDef
<b>BSW Description</b>	
Provide the size of a RoutineControl parameter in bytes	
<b>Template Description</b>	
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.	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
"Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00187

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal



BSW Parameter		BSW Type
DcmDspRoutineSignalEndianness		EcucEnumerationParamDef
BSW Description		
Defines the endianness of the data belonging to a Routine Out Signal for Stop subfunction.		
Template Description		
This attribute specifies the byte order of the base type.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder		
Mapping Rule		Mapping Type
baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response		full
Mapping Status		Mapping ID
valid		up_Dcm_00184

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal	
BSW Parameter		BSW Type
DcmDspRoutineSignalPos		EcucIntegerParamDef
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		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.bitOffset		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00191

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType	
BSW Parameter		BSW Type
BOOLEAN		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is boolean.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
Mapping Rule		Mapping Type

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response	full
baseTypeEncoding = BOOLEAN baseTypeSize = 1	
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00195

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType	
BSW Parameter		BSW Type
SINT16		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is sint16.		
Template Description		
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.		
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
Mapping Rule		Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response		full
baseTypeEncoding = 2C baseTypeSize = 16		
Mapping Status		Mapping ID
valid		up_Dcm_00159

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType	
BSW Parameter		BSW Type
SINT16_N		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is sint16 array.		
Template Description		

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.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 ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00159

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
SINT32	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is sint32.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response  baseTypeEncoding = 2C baseTypeSize = 32	full

Mapping Status	Mapping ID
valid	up_Dcm_00162

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	BSW Type
SINT32_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint32 array.	
Template Description	
<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p> <p><b>DiagnosticDataElement.maxNumberOfElements:</b> 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.</p> <p><b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.</p>	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	Mapping Type
<p>referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response</p> <p>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)</p>	full
Mapping Status	Mapping ID
valid	up_Dcm_00162

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	BSW Type
SINT8	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint8.	
Template Description	

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response  baseTypeEncoding = 2C baseTypeSize = 8	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00203

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
SINT8_N	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is sint8 array.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response  baseTypeEncoding = 2C baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either doesnot exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full

Mapping Status	Mapping ID
valid	up_Dcm_00203

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	BSW Type
UINT16	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint16.	
Template Description	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response  baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16	full
Mapping Status	Mapping ID
valid	up_Dcm_00209

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	BSW Type
UINT16_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint16 array.	
Template Description	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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	

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response  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 ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00209

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT32	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is uint32.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response  baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00212

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT32_N	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is uint32 array.	
<b>Template Description</b>	

<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response  baseTypeEncoding = NONE, UTF-32 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
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00212

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
<b>BSW Parameter</b>	<b>BSW Type</b>
UINT8	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Type of the signal is uint8.	
<b>Template Description</b>	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>M2 Parameter</b>	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>



referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response	full
baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8	
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00170

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	BSW Type
UINT8_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint8 array.	
Template Description	
<p><b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.</p> <p><b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.</p> <p><b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.</p> <p><b>DiagnosticDataElement.maxNumberOfElements:</b> 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.</p>	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response	full
baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP 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.fixedSize (cf. TPS_DEXT_01001)	
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00170

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
VARIABLE_LENGTH	EcucEnumerationLiteralDef
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	
<b>BaseTypeDirectDefinition.baseTypeEncoding:</b> This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.	
<b>BaseTypeDirectDefinition.baseTypeSize:</b> Describes the length of the data type specified in the container in bits.	
<b>DiagnosticDataElement.maxNumberOfElements:</b> 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.	
<b>DiagnosticDataElement.arraySizeSemantics:</b> This attribute controls the meaning of the value of the array size.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response  baseTypeEncoding = NONE baseTypeSize = 8 arraySizeSemantics = variableSize 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	Mapping ID
valid	up_Dcm_00173

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	BSW Type
DcmDspSecurityADRSIZE	EcucIntegerParamDef
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	
DiagnosticExtract::Dcm::DiagnosticSecurityLevel.accessDataRecordSize	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	Mapping ID

valid	up_Dcm_00241
-------	--------------

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow	
BSW Parameter	BSW Type	
DcmDspSecurityAttemptCounterEnabled	EcucBooleanParamDef	
BSW Description		
<p>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.</p>		
Template Description		
<p>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.</p>		
M2 Parameter		
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00293

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow	
BSW Parameter	BSW Type	
DcmDspSecurityCompareKeyFnc	EcucFunctionNameDef	
BSW Description		
<p>Function name to request the result of a key comparison.</p> <p>Parameter is only relevant if DcmDspSecurityUsePort=="USE_ ASYNCH_FNC". This parameter is related to the interface Xxx_CompareKey.</p>		
Template Description		
<p>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.</p>		
M2 Parameter		
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00240

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow	
BSW Parameter	BSW Type	
DcmDspSecurityDelayTime	EcucFloatParamDef	
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.	
<b>Template Description</b>	
This represents the delay time after a failed security access. Unit: second.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticSecurityLevel.securityDelayTime	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00238

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspSecurityDelayTimeOnBoot	EcucFloatParamDef
<b>BSW Description</b>	
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.	
<b>Template Description</b>	
Start delay timer on power on in seconds.	
This delay indicates the time at ECU boot power-on time where the Dcm remains in the default session and does not accept a security access.	
<b>M2 Parameter</b>	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.securityDelayTimeOnBoot	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00237

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspSecurityGetAttemptCounterFnc	EcucFunctionNameDef
<b>BSW Description</b>	
Function name to request the value of an attempt counter. This parameter is related to the interface Xxx_GetSecurityAttemptCounter.	
<b>Template Description</b>	
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.	
<b>M2 Parameter</b>	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full

Mapping Status	Mapping ID
valid	up_Dcm_00291

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow	
BSW Parameter	BSW Type	
DcmDspSecurityGetSeedFnc	EcucFunctionNameDef	
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	DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00239	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow	
BSW Parameter	BSW Type	
DcmDspSecurityKeySize	EcucIntegerParamDef	
BSW Description	size of the security key (in Bytes).	
Template Description	This represents the size of the security key. Unit: byte.	
M2 Parameter	DiagnosticExtract::Dcm::DiagnosticSecurityLevel.keySize	
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00236	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow	
BSW Parameter	BSW Type	
DcmDspSecurityLevel	EcucIntegerParamDef	
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	
<b>Template Description</b>	
This would be 0x01, 0x03, 0x05, ...	
The sendKey id can be computed by adding 1 to the requestSeedId	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::SecurityAccess::DiagnosticSecurityAccess.requestSeedId	
<b>Mapping Rule</b>	<b>Mapping Type</b>
DcmDspSecurityLevel=(requestSeedId+1)/2	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00235

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspSecurityNumAttDelay	EcucIntegerParamDef
<b>BSW Description</b>	
Number of failed security accesses after which the delay time is activated	
<b>Template Description</b>	
This represents the number of failed security accesses after which the delay time is activated.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticSecurityLevel.numFailedSecurityAccess	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00234

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspSecuritySeedSize	EcucIntegerParamDef
<b>BSW Description</b>	
size of the security seed (in Bytes).	
<b>Template Description</b>	
This represents the size of the security seed. Unit: byte.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticSecurityLevel.seedSize	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00233

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow	
BSW Parameter		BSW Type
DcmDspSecuritySetAttemptCounterFnc		EcucFunctionNameDef
BSW Description		
Function name to set the value of an attempt counter. This parameter is related to the interface Xxx_SetSecurityAttemptCounter.		
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		
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00292

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow/DcmDspSecurityUsePort	
BSW Parameter		BSW Type
USE_ASYNCH_CLIENT_SERVER		EcucEnumerationLiteralDef
BSW Description		
The DCM will access the data using an R-Port requiring a asynchronous ClientServerInterface SecurityAccess_{SecurityLevel}.		
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 rootSoftwareComposition)		
M2 Parameter		
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedSwcServiceDependencyInSystem		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00226

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow/DcmDspSecurityUsePort	
BSW Parameter		BSW Type
USE_ASYNCH_FNC		EcucEnumerationLiteralDef
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 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	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	Mapping ID
valid	up_Dcm_00225

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow/DcmDspSessionForBoot	
BSW Parameter		BSW Type
DCM_NO_BOOT		EcucEnumerationLiteralDef
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		
DiagnosticExtract::Dcm::DiagnosticJumpToBootLoaderEnum.noBoot		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00231	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow/DcmDspSessionForBoot	
BSW Parameter		BSW Type
DCM_OEM_BOOT		EcucEnumerationLiteralDef
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		
DiagnosticExtract::Dcm::DiagnosticJumpToBootLoaderEnum.oemBoot		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00227	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow/DcmDspSessionForBoot	
BSW Parameter		BSW Type
DCM_OEM_BOOT_RESPAPP		EcucEnumerationLiteralDef



<b>BSW Description</b>	
This diagnostic session allows to jump to OEM Bootloader and application sends final response.	
<b>Template Description</b>	
This diagnostic session allows to jump to OEM Bootloader and application sends final response.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticJumpToBootLoaderEnum.oemBootRespApp	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00228

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow/DcmDspSessionForBoot	
<b>BSW Parameter</b>		<b>BSW Type</b>
DCM_SYS_BOOT		EcucEnumerationLiteralDef
<b>BSW Description</b>		
This diagnostic session allows to jump to System Supplier Bootloader and bootloader sends final response.		
<b>Template Description</b>		
This diagnostic session allows to jump to System Supplier Bootloader. In this case the bootloader send the final response.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticJumpToBootLoaderEnum.systemSupplierBoot		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
1:1 mapping	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid	up_Dcm_00229	

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow/DcmDspSessionForBoot	
<b>BSW Parameter</b>		<b>BSW Type</b>
DCM_SYS_BOOT_RESPAPP		EcucEnumerationLiteralDef
<b>BSW Description</b>		
This diagnostic session allows to jump to System Supplier Bootloader and application sends final response.		
<b>Template Description</b>		
This diagnostic session allows to jump to System Supplier Bootloader and application sends final response.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticJumpToBootLoaderEnum.systemSupplierBootRespApp		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
1:1 mapping	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid	up_Dcm_00230	

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow	
<b>BSW Parameter</b>		<b>BSW Type</b>

DcmDspSessionLevel	EcucIntegerParamDef
<b>BSW Description</b>	
subFunction value of the DiagnosticSession.	
0, 127 and all values above 127 are reserved by ISO	
<b>Template Description</b>	
This is the numerical identifier used to identify the DiagnosticSession in the scope of diagnostic workflow	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticSession.id	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00232

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow	
<b>BSW Parameter</b>	<b>BSW Type</b>	
DcmDspSessionP2ServerMax	EcucFloatParamDef	
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
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.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticSession.p2ServerMax		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
1:1 mapping	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid	up_Dcm_00251	

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow	
<b>BSW Parameter</b>	<b>BSW Type</b>	
DcmDspSessionP2StarServerMax	EcucFloatParamDef	
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
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.		

<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticSession.p2StarServerMax	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00252

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspVehInfo		EcucParamConfContainerDef
<b>BSW Description</b>		
This container contains the configuration (parameters) for one single VehicleInfoType of service \$09		
<b>Template Description</b>		
This meta-class represents the ability to model an instance of the OBD mode 0x09 service.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::ObdService::Mode_0x09_RequestVehicleInformation::DiagnosticRequestVehicleInfo		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
1:1 mapping	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid		

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspVehInfoData		EcucParamConfContainerDef
<b>BSW Description</b>		
Data Item of an InfoType; ShortName is post-fix of the port interface name.		
<b>Template Description</b>		
This meta-class represents the ability to describe a concrete piece of data to be taken into account for diagnostic purposes.		
<b>M2 Parameter</b>		
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
The value shall be taken from DiagnosticRequestVehicleInfo.infoType.dataElement.dataElement.	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid		

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDspVehInfoDataOrder		EcucIntegerParamDef
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
This represents the bitOffset of the DiagnosticParameter		

<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.bitOffset	
<b>Mapping Rule</b>	<b>Mapping Type</b>
Value shall be taken from DiagnosticRequestVehicleInfo.infoType.dataElement.bitOffset.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspVehInfoDataReadFnc	EcucFunctionNameDef
<b>BSW Description</b>	
Callout function name for reading InfoType data item. Only required in case parameter 'DcmDspVehInfoDataUsePort' is set to 'false'	
<b>Template Description</b>	
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.	
<b>M2 Parameter</b>	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency	
<b>Mapping Rule</b>	<b>Mapping Type</b>
The BswServiceDependency should have aRoleBasedBswModuleEntryAssignment that in turn has attribute role set to Xxx_GetInfotypeValueData and points to a BswModuleEntry.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspVehInfoDataSize	EcucIntegerParamDef
<b>BSW Description</b>	
Size in bytes of the InfoType data item.	
<b>Template Description</b>	
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.	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
<b>Mapping Rule</b>	<b>Mapping Type</b>
Value shall be taken from DiagnosticRequestVehicleInfo.infoType.dataElement.dataElement.maxNumberOfElements.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspVehInfoDataUsePort	EcucBooleanParamDef
<b>BSW Description</b>	

<p>When this parameter is set to true the DCM will access the Data using an R-Port requiring a PortInterface IfnotypeServices_{VehInfoData}.</p> <p>The R-Port is named InfotypeServices_{VehInfoData} where {VEHINFODATA} is the name of the container DcmDspVehInfoData. In that case, the DcmDspVehInfoDataReadFnc is ignored and the RTE APIs are used.</p> <p>When this parameter is set to false, the DCM calls the function defined in DcmDspVehInfoDataReadFnc.</p>	
<b>Template Description</b>	
<p><b>DiagnosticServiceSwMapping.mappedBswServiceDependency:</b> 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.</p> <p><b>DiagnosticServiceSwMapping.mappedSwcServiceDependencyInSystem:</b> This represents the ability to point into the component hierarchy (under possible consideration of the rootSoftwareComposition)</p>	
<b>M2 Parameter</b>	
<p>DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency, DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedSwcServiceDependencyInSystem</p>	
<b>Mapping Rule</b>	<b>Mapping Type</b>
<p>Shall be set to TRUE if the reference DiagnosticServiceSwMapping.mappedSwcServiceDependency exists.</p> <p>Shall be set to FALSE if the reference DiagnosticServiceSwMapping.mappedBswServiceDependency exists.</p>	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmDspVehInfoInfoType	EcucIntegerParamDef
<b>BSW Description</b>	
value of InfoType.	
Within each DcmConfigSet all DcmDspVehInfoInfoType values shall be unique.	
<b>Template Description</b>	
<p><b>ObdInfoServiceNeeds.infoType:</b> The InfoType according to ISO 15031-5</p> <p><b>DiagnosticInfoType.id:</b> This attribute represents the value of InfoType (see SAE J1979-DA).</p>	
<b>M2 Parameter</b>	
<p>CommonStructure::ServiceNeeds::ObdInfoServiceNeeds.infoType, DiagnosticExtract::CommonDiagnostics::DiagnosticInfoType.id</p>	
<b>Mapping Rule</b>	<b>Mapping Type</b>
If DiagnosticRequestVehicleInfo, us DiagnosticRequestVehicleInfo.infoType.id.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00029

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions	
BSW Parameter	BSW Type	
DcmModeCondition	EcucParamConfContainerDef	
BSW Description		
<p>This container contains the configuration of a mode condition or an environmental conditions which can be used as argument in DcmModeRules.</p> <p>One DcmModeCondition shall contain either one DcmSwcModeRef or one DcmBswModeRef or one DcmSwcSRDataElementRef.</p> <p>Please note that the Dcm acts as well as mode manager. Therefore the references DcmSwcModeRef or one DcmBswModeRef might point to provided ModeDeclarationGroupPrototypes of the Dcm itself as well as to provided ModeDeclarationGroupPrototypes of other Bsw Modules or software components.</p> <p>In case of a configured DcmSwcModeRef or DcmBswModeRef only the DcmConditionType DCM_EQUALS or DCM_EQUALS_NOT are applicable.</p> <p>In case of DcmSwcSRDataElementRef all literals of DcmConditionType are possible.</p>		
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 DiagnosticCompareCondition.compareType.		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvCompareCondition		
Mapping Rule		Mapping Type
Depending on the reference a DcmModeCondition is mapped to a Diagnostic EnvModeCondition if only one reference is present and reference is a DcmSwc ModeRef or a DcmBswModeRef. If two references are present, a DcmSwc SRDataElementRef and a DcmSwcSRDataElementValueRef, then DcmMode Condition is mapped to a DiagnosticEnvDataCondition.		full
Mapping Status		Mapping ID
valid		up_Dcm_00271

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition	
BSW Parameter	BSW Type	
DcmBswModeRef	EcucInstanceReferenceDef	
BSW Description		
<p>This parameter references a mode of a ModeDeclarationGroupPrototype provided by a Basic Software Module used for the condition.</p> <p>Please note that such ModeDeclarationGroupPrototype are owned by a Basic Software Module Description in the role providedModeGroup.</p>		
Template Description		
This reference represents both the ModeDeclarationGroupPrototype and the ModeDeclaration relevant for the mode comparison.		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::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 DiagnosticEnvironmentalCondition as the DiagnosticEnvModeCondition, in which the target of the reference DiagnosticEnvModeCondition.modeElement points to the this DiagnosticEnvModeElement.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00273

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType	
BSW Parameter	BSW Type	
DCM_EQUALS	EcucEnumerationLiteralDef	
BSW Description		
Template Description		
<b>DiagnosticEnvConditionFormula.op:</b> This attribute represents the concrete operator (supported operators: and, or) of the condition formula.		
<b>DiagnosticCompareTypeEnum.isEqual:</b> equal		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isEqual		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00265

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType	
BSW Parameter	BSW Type	
DCM_EQUALS_NOT	EcucEnumerationLiteralDef	
BSW Description		
Template Description		
<b>DiagnosticEnvConditionFormula.op:</b> This attribute represents the concrete operator (supported operators: and, or) of the condition formula.		
<b>DiagnosticCompareTypeEnum.isNotEqual:</b> not equal		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isNotEqual		

<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00266

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType	
<b>BSW Parameter</b>		<b>BSW Type</b>
DCM_GREATER_OR_EQUAL		EcucEnumerationLiteralDef
<b>BSW Description</b>		
<b>Template Description</b>		
<p><b>DiagnosticEnvConditionFormula.op:</b> This attribute represents the concrete operator (supported operators: and, or) of the condition formula.</p> <p><b>DiagnosticCompareTypeEnum.isGreaterOrEqual:</b> greater than or equal</p>		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isGreaterOrEqual		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00269

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType	
<b>BSW Parameter</b>		<b>BSW Type</b>
DCM_GREATER_THAN		EcucEnumerationLiteralDef
<b>BSW Description</b>		
<b>Template Description</b>		
<p><b>DiagnosticEnvConditionFormula.op:</b> This attribute represents the concrete operator (supported operators: and, or) of the condition formula.</p> <p><b>DiagnosticCompareTypeEnum.isGreaterThan:</b> greater than</p>		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isGreaterThan		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00267



BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType	
BSW Parameter		BSW Type
DCM_LESS_OR_EQUAL		EcucEnumerationLiteralDef
BSW Description		
<p><b>Template Description</b></p> <p><b>DiagnosticEnvConditionFormula.op:</b> This attribute represents the concrete operator (supported operators: and, or) of the condition formula.</p> <p><b>DiagnosticCompareTypeEnum.isLessOrEqual:</b> less than or equal</p> <p><b>M2 Parameter</b> DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isLessOrEqual</p>		
Mapping Rule		Mapping Type
		full
Mapping Status		Mapping ID
valid		up_Dcm_00270

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType	
BSW Parameter		BSW Type
DCM_LESS_THAN		EcucEnumerationLiteralDef
BSW Description		
<p><b>Template Description</b></p> <p><b>DiagnosticEnvConditionFormula.op:</b> This attribute represents the concrete operator (supported operators: and, or) of the condition formula.</p> <p><b>DiagnosticCompareTypeEnum.isLessThan:</b> less than</p> <p><b>M2 Parameter</b> DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isLessThan</p>		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00268

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmSwcDataElementValue/DcmSwcDataElementArray/DcmSwcDataElementArrayElement	

<b>BSW Parameter</b>		<b>BSW Type</b>	
DcmSwcDataElementArrayElementIndex		EcuIntegerParamDef	
<b>BSW Description</b>			
Index to an element of the compare value array.			
<b>Template Description</b>			
This attribute represents a fixed compare value taken to evaluate the compare condition.			
<b>M2 Parameter</b>			
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvDataCondition.compareValue			
<b>Mapping Rule</b>		<b>Mapping Type</b>	
DiagnosticDataElement referenced by the DiagnosticEnvDataCondition is an array.		full	
<b>Mapping Status</b>		<b>Mapping ID</b>	
valid		up_Dcm_00298	

<b>BSW Module</b>	<b>BSW Context</b>		
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmSwcDataElementValue/DcmSwcDataElementArray/DcmSwcDataElementArrayElement		
<b>BSW Parameter</b>		<b>BSW Type</b>	
DcmSwcDataElementArrayElementValue		EcuIntegerParamDef	
<b>BSW Description</b>			
Value of an array element compare value.			
<b>Template Description</b>			
This attribute represents a fixed compare value taken to evaluate the compare condition.			
<b>M2 Parameter</b>			
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvDataCondition.compareValue			
<b>Mapping Rule</b>		<b>Mapping Type</b>	
DiagnosticDataElement referenced by the DiagnosticEnvDataCondition is an array.		full	
<b>Mapping Status</b>		<b>Mapping ID</b>	
valid		up_Dcm_00299	

<b>BSW Module</b>	<b>BSW Context</b>		
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmSwcDataElementValue/DcmSwcDataElementPrimitive		
<b>BSW Parameter</b>		<b>BSW Type</b>	
DcmSwcDataElementPrimitiveValue		EcuIntegerParamDef	
<b>BSW Description</b>			
Primitive compare value.			
<b>Template Description</b>			
This attribute represents a fixed compare value taken to evaluate the compare condition.			
<b>M2 Parameter</b>			
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvDataCondition.compareValue			
<b>Mapping Rule</b>		<b>Mapping Type</b>	
DiagnosticDataElement referenced by the DiagnosticEnvDataCondition is primitive.		full	
<b>Mapping Status</b>		<b>Mapping ID</b>	
valid		up_Dcm_00297	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition	
BSW Parameter		BSW Type
DcmSwcModeRef		EcucInstanceReferenceDef
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		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvModeCondition.modeElement		
Mapping Rule		Mapping Type
For DcmModeRef a new DiagnosticEnvSwcModeElement is used, pointing to the ModeDeclaration via PModelInSystemInstanceRef. This new DiagnosticEnvModeElement shall be aggregated by the same DiagnosticEnvironmentalCondition as the DiagnosticEnvModeCondition, in which the target of the reference DiagnosticEnvModeCondition.modeElement points to the this DiagnosticEnvModeElement.		full
Mapping Status		Mapping ID
valid		up_Dcm_00272

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition	
BSW Parameter		BSW Type
DcmSwcSRDataElementRef		EcucChoiceReferenceDef
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		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvDataCondition.dataElement		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00274

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions	
BSW Parameter		BSW Type
DcmModeRule		EcucParamConfContainerDef
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		

<p>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.</p> <p>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.</p>	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula	
<b>Mapping Rule</b>	<b>Mapping Type</b>
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 DiagnosticEnvironmentalCondition shall be created with DiagnosticEnvironmentalCondition.formula containing a DiagnosticEnvConditionFormula. In both cases, if no DcmLogicalOperator is present in this DcmModeRule, then DiagnosticEnvConditionFormula shall be set to DiagnosticLogicalOperatorEnum.logicalAnd.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00259

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeRule
<b>BSW Parameter</b>	<b>BSW Type</b>
DcmArgumentRef	EcucChoiceReferenceDef
<b>BSW Description</b>	
This is a choice reference either to a mode condition or a an other mode rule serving as sub-expression.	
<b>Template Description</b>	
A DiagnosticEnvConditionFormulaPart can either be a atomic condition, e.g. a DiagnosticEnvCompareCondition, or a DiagnosticEnvConditionFormula, again, which allows arbitrary nesting.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormulaPart	
<b>Mapping Rule</b>	<b>Mapping Type</b>
Depending on the destination, one DcmArgumentRef is mapped to a DiagnosticEnvConditionFormula if "destination" is a DcmModeRule, and to a DiagnosticEnvCompareCondition, if "destination" is a DcmModeCondition. The order of the aggregation of the DiagnosticEnvConditionFormulaParts shall correspond to the ordering of the index of the DcmArgumentRefs.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00264

<b>BSW Module</b>	<b>BSW Context</b>
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeRule/DcmLogicalOperator
<b>BSW Parameter</b>	<b>BSW Type</b>
DCM_AND	EcucEnumerationLiteralDef
<b>BSW Description</b>	

Template Description	
<b>DiagnosticEnvConditionFormula.op:</b> This attribute represents the concrete operator (supported operators: and, or) of the condition formula.	
<b>DiagnosticLogicalOperatorEnum.logicalAnd:</b> Logical AND	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticLogicalOperatorEnum.logicalAnd	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	Mapping ID
valid	up_Dcm_00261

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeRule/DcmLogicalOperator	
BSW Parameter		BSW Type
DCM_OR		EcucEnumerationLiteralDef
BSW Description		
Template Description		
<b>DiagnosticEnvConditionFormula.op:</b> This attribute represents the concrete operator (supported operators: and, or) of the condition formula.		
<b>DiagnosticLogicalOperatorEnum.logicalOr:</b> Logical OR		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticLogicalOperatorEnum.logicalOr		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00262	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeRule	
BSW Parameter		BSW Type
DcmModeRuleNrcValue		EcucIntegerParamDef
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		

DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvCondition Formula.nrcValue	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00263

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmGeneral	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmDDDIDStorage		EcucBooleanParamDef
<b>BSW Description</b>		
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		
<b>Template Description</b>		
This configuration switch defines whether DDDID definition is handled as non-volatile information or not.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier::DiagnosticDynamicallyDefineDataIdentifierClass.configurationHandling		
<b>Mapping Rule</b>		<b>Mapping Type</b>
volatile -> false, nonVolatile -> true		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00253

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmGeneral	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmRespondAllRequest		EcucBooleanParamDef
<b>BSW Description</b>		
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).		
<b>Template Description</b>		
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).		
<b>M2 Parameter</b>		
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.responseOnAllRequest Sids		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00249

<b>BSW Module</b>	<b>BSW Context</b>	
Dcm	Dcm/DcmGeneral	
<b>BSW Parameter</b>		<b>BSW Type</b>
DcmVinRef		EcucReferenceDef
<b>BSW Description</b>		

Reference to the Did containing the VIN Information.	
This parameter is needed for function Dcm_GetVin	
<b>Template Description</b>	
This meta-class represents the ability to model a diagnostic data identifier (DID) that is fully specified regarding the payload at configuration-time.	
<b>M2 Parameter</b>	
DiagnosticExtract::CommonDiagnostics::DiagnosticDataIdentifier	
<b>Mapping Rule</b>	<b>Mapping Type</b>
This reference shall only apply to a DiagnosticDataIdentifier where the attribute representsVin is set to true.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00248

### E.3 Dem

The Dem upstream mapping has been removed from publication in response to the existence of various deficiencies. It shall be fixed and completed in the next release.

### E.4 Fim

<b>BSW Module</b>	<b>BSW Context</b>	
FiM	FiM/FiMConfigSet	
<b>BSW Parameter</b>		<b>BSW Type</b>
FiMFID		EcucParamConfContainerDef
<b>BSW Description</b>		
This container includes symbolic names of all FIDs.		
<b>Template Description</b>		
<b>FunctionInhibitionNeeds:</b> 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.		
<b>DiagnosticFunctionIdentifier:</b> This meta-class represents a diagnostic function identifier (a.k.a. FID).		
<b>M2 Parameter</b>		
CommonStructure::ServiceNeeds::FunctionInhibitionNeeds DiagnosticExtract::Fim::DiagnosticFunctionIdentifier		
<b>Mapping Rule</b>		<b>Mapping Type</b>
In case the owner of the FunctionInhibitionNeeds is a BSW module then the FiMFID.shortName= {capitalizedMip}_{ServiceDependency.symbolicName Props.symbol}.		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_FiM_00001

<b>BSW Module</b>	<b>BSW Context</b>	
FiM	FiM/FiMConfigSet	
<b>BSW Parameter</b>		<b>BSW Type</b>
FiMinhibitionConfiguration		EcucParamConfContainerDef

<b>BSW Description</b>	
This container includes all configuration parameters concerning the relationship between event and FID.	
<b>Template Description</b>	
This meta-class represents the ability to define the inhibition of a specific function identifier within the Fim configuration.	
<b>M2 Parameter</b>	
DiagnosticExtract::Fim::DiagnosticFunctionIdentifierInhibit	
<b>Mapping Rule</b>	<b>Mapping Type</b>
	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
FiM	FiM/FiMConfigSet/FiMinhibitionConfiguration
<b>BSW Parameter</b>	<b>BSW Type</b>
FiMinhEventRef	EcucReferenceDef
<b>BSW Description</b>	
Selection of an single DEM Event.	
<b>Template Description</b>	
This represents the alias event applicable for the referencing inhibition source.	
<b>M2 Parameter</b>	
DiagnosticExtract::Fim::DiagnosticFunctionInhibitSource.event	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
FiM	FiM/FiMConfigSet/FiMinhibitionConfiguration
<b>BSW Parameter</b>	<b>BSW Type</b>
FiMinhInhibitionMask	EcucEnumerationParamDef
<b>BSW Description</b>	
The configuration parameter is used to specify the inhibition mask for an event - FID relation.	
<b>Template Description</b>	
This represents the value of the inhibition mask behavior.	
<b>M2 Parameter</b>	
DiagnosticExtract::Fim::DiagnosticFunctionIdentifierInhibit.inhibitionMask	
<b>Mapping Rule</b>	<b>Mapping Type</b>
	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
FiM	FiM/FiMConfigSet/FiMinhibitionConfiguration/FiMinhInhibitionMask
<b>BSW Parameter</b>	<b>BSW Type</b>
FIM_LAST_FAILED	EcucEnumerationLiteralDef
<b>BSW Description</b>	
Last Failed - DEM_UDS_STATUS_TF flag of Dem Eventstatus is set Use case: Re-configuration, avoiding follow-up errors	



Template Description	
This represents the inhibition mask behavior "last failed".	
M2 Parameter	
DiagnosticExtract::Fim::DiagnosticInhibitionMaskEnum.lastFailed	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	Mapping ID
valid	

BSW Module	BSW Context
FiM	FiM/FiMConfigSet/FiMinhibitionConfiguration/FiMinhInhibitionMask
BSW Parameter	BSW Type
FIM_NOT_TESTED	EcucEnumerationLiteralDef
BSW Description	
Not Tested this cycle - DEM_UDS_STATUS_TNCTOC flag of Dem Eventstatus is set. Use case: Scheduling of monitors.	
Template Description	
This represents the inhibition mask behavior "not tested".	
M2 Parameter	
DiagnosticExtract::Fim::DiagnosticInhibitionMaskEnum.notTested	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	Mapping ID
valid	

BSW Module	BSW Context
FiM	FiM/FiMConfigSet/FiMinhibitionConfiguration/FiMinhInhibitionMask
BSW Parameter	BSW Type
FIM_TESTED	EcucEnumerationLiteralDef
BSW Description	
Tested - DEM_UDS_STATUS_TNCTOC flag of Dem Eventstatus is not set. Use case: Self deactivation, check during driving cycle.	
Template Description	
This represents the inhibition mask behavior "tested".	
M2 Parameter	
DiagnosticExtract::Fim::DiagnosticInhibitionMaskEnum.tested	
Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	Mapping ID
valid	

BSW Module	BSW Context
FiM	FiM/FiMConfigSet/FiMinhibitionConfiguration/FiMinhInhibitionMask
BSW Parameter	BSW Type
FIM_TESTED_AND_FAILED	EcucEnumerationLiteralDef
BSW Description	
Tested and Failed - DEM_UDS_STATUS_TF flag of Dem Eventstatus is set and DEM_UDS_STATUS_TNCTOC flag is not set Use case: Avoiding deadlocks, repeated monitoring.	
Template Description	

This represents the inhibition mask behavior "tested and failed".	
<b>M2 Parameter</b>	
DiagnosticExtract::Fim::DiagnosticInhibitionMaskEnum.testedAndFailed	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
FiM	FiM/FiMConfigSet
<b>BSW Parameter</b>	<b>BSW Type</b>
FiMSummaryEvent	EcucParamConfContainerDef
<b>BSW Description</b>	
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.	
<b>Template Description</b>	
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.	
<b>M2 Parameter</b>	
DiagnosticExtract::Fim::DiagnosticFimEventGroup	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

## E.5 J1939 Dcm

<b>BSW Module</b>	<b>BSW Context</b>
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmChannel
<b>BSW Parameter</b>	<b>BSW Type</b>
J1939DcmBusType	EcucEnumerationParamDef
<b>BSW Description</b>	
Identifies the communication port	
<b>Template Description</b>	
This represents the network ID for the J1939 cluster.	
<b>M2 Parameter</b>	
SystemTemplate::Fibex::Fibex4Can::CanTopology::J1939Cluster.networkId	
<b>Mapping Rule</b>	<b>Mapping Type</b>
value 1 maps to J1939DCM_J1939_NETWORK_1 value 2 maps to J1939DCM_J1939_NETWORK_2 value 3 maps to J1939DCM_J1939_NETWORK_3 value 4 maps to J1939DCM_J1939_NETWORK_4	full

Mapping Status	Mapping ID
valid	

BSW Module	BSW Context	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmDspExternalSRDataElementClass/J1939DcmDataElementInstance	
BSW Parameter		BSW Type
J1939DcmDataElementInstanceRef		EcucInstanceReferenceDef
BSW Description		
<p>Instance Reference to the primitive 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 if the AutosarDataPrototype is typed with a ImplementationDataType of category VALUE or TYPE_REFERENCE that in turn boils down to VALUE</p>		
Template Description		
<p>This represents the dataElement in the application software that is accessed for diagnostic purpose. This role is applicable on the classic platform.</p>		
M2 Parameter		
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping.mappedDataElement		
Mapping Rule		Mapping Type
DiagnosticServiceDataMapping maps to a primitive data.		full
Mapping Status		Mapping ID
valid		up_Dcm_00100

BSW Module	BSW Context	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmDspExternalSRDataElementClass/J1939DcmSubElementInDataElementInstance	
BSW Parameter		BSW Type
J1939DcmSubElementInDataElementInstanceRef		EcucInstanceReferenceDef
BSW Description		
<p>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.</p>		
Template Description		
<p>This represents the dataElement in the application software that is accessed for diagnostic purpose. This role is applicable on the classic platform.</p>		
M2 Parameter		
DiagnosticExtract::ServiceMapping::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		Mapping ID
valid		up_Dcm_00101

BSW Module	BSW Context
------------	-------------

J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmDspExternalSRDataElementClass/J1939DcmSubElementInImplDataElementInstance	
<b>BSW Parameter</b>		<b>BSW Type</b>
J1939DcmSubElementInImplDataElementInstanceRef		EcucInstanceReferenceDef
<b>BSW Description</b>		
Instance Reference to the primitive sub-element (at any level) of composite data in a port which shall be read. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ImplementationDataType of category STRUCTURE or ARRAY. Please note that in case of ARRAY the index attribute in the target reference has to be set to select a single array element.		
<b>Template Description</b>		
This represents the dataElement in the application software that is accessed for diagnostic purpose. This role is applicable on the classic platform.		
<b>M2 Parameter</b>		
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping.mappedDataElement		
<b>Mapping Rule</b>		<b>Mapping Type</b>
DiagnosticServiceDataMapping maps to a primitive element within a composite data, where the AutosarDataPrototype is typed with a ApplicationCompositeDataType ImplementationDataType of category STRUCTURE or ARRAY.		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00102

<b>BSW Module</b>	<b>BSW Context</b>	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmNode	
<b>BSW Parameter</b>		<b>BSW Type</b>
J1939DcmServiceOnlyDTCsMemoryDestinationRef		EcucReferenceDef
<b>BSW Description</b>		
Reference to the user defined memory used for the Service Only DTCs handled by DM53, DM54, and DM55.		
<b>Template Description</b>		
The event destination assigns events to none, one or multiple origins.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticTroubleCodeProps.memoryDestination		
<b>Mapping Rule</b>		<b>Mapping Type</b>
This upstream mapping shall only exist for a DiagnosticTroubleCodeJ1939 with attribute kind set to serviceOnly that references the DiagnosticTroubleCode Props that owns the memoryDestination.		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_J1939Dcm_00005

<b>BSW Module</b>	<b>BSW Context</b>	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions	
<b>BSW Parameter</b>		<b>BSW Type</b>
J1939DcmModeCondition		EcucParamConfContainerDef
<b>BSW Description</b>		

<p>This container contains the configuration of a mode condition or an environmental conditions which can be used as argument in J1939DcmModeRules.</p> <p>One J1939DcmModeCondition shall contain either one J1939DcmSwcModeRef or one J1939DcmBswModeRef or one J1939DcmSwcSRDataElementRef.</p> <p>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.</p> <p>In case of a configured J1939DcmSwcModeRef or J1939DcmBswModeRef only the J1939DcmConditionType J1939DCM_EQUALS or J1939DCM_EQUALS_NOT are applicable.</p> <p>In case of J1939DcmSwcSRDataElementRef all literals of J1939DcmConditionType are possible.</p>	
<b>Template Description</b>	
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.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvCompareCondition	
<b>Mapping Rule</b>	<b>Mapping Type</b>
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
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition
<b>BSW Parameter</b>	<b>BSW Type</b>
J1939DcmBswModeRef	EcucInstanceReferenceDef
<b>BSW Description</b>	
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.	
<b>Template Description</b>	
This reference represents both the ModeDeclarationGroupPrototype and the ModeDeclaration relevant for the mode comparison.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvModeCondition.modeElement	
<b>Mapping Rule</b>	<b>Mapping Type</b>

For DcmModeRef a new DiagnosticEnvBswModeElement is used, pointing to the ModeDeclaration via ModeInModuleDescriptionInstanceRef. This new DiagnosticEnvModeElement shall be aggregated by the same DiagnosticEnvironmentalCondition as the DiagnosticEnvModeCondition, in which the target of the reference DiagnosticEnvModeCondition.modeElement points to the this DiagnosticEnvModeElement.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00273

BSW Module	BSW Context	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType	
BSW Parameter	BSW Type	
J1939DCM_EQUALS	EcucEnumerationLiteralDef	
BSW Description		
<b>Template Description</b>		
<b>DiagnosticEnvConditionFormula.op:</b> This attribute represents the concrete operator (supported operators: and, or) of the condition formula.		
<b>DiagnosticCompareTypeEnum.isEqual:</b> equal		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isEqual		
Mapping Rule	Mapping Type	
1:1 mapping	full	
Mapping Status	Mapping ID	
valid	up_Dcm_00265	

BSW Module	BSW Context	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType	
BSW Parameter	BSW Type	
J1939DCM_EQUALS_NOT	EcucEnumerationLiteralDef	
BSW Description		
<b>Template Description</b>		
<b>DiagnosticEnvConditionFormula.op:</b> This attribute represents the concrete operator (supported operators: and, or) of the condition formula.		
<b>DiagnosticCompareTypeEnum.isNotEqual:</b> not equal		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isNotEqual		

<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00266

<b>BSW Module</b>	<b>BSW Context</b>	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType	
<b>BSW Parameter</b>		<b>BSW Type</b>
J1939DCM_GREATER_OR_EQUAL		EcucEnumerationLiteralDef
<b>BSW Description</b>		
<b>Template Description</b>		
<p><b>DiagnosticEnvConditionFormula.op:</b> This attribute represents the concrete operator (supported operators: and, or) of the condition formula.</p> <p><b>DiagnosticCompareTypeEnum.isGreaterOrEqual:</b> greater than or equal</p>		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isGreaterOrEqual		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00269

<b>BSW Module</b>	<b>BSW Context</b>	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType	
<b>BSW Parameter</b>		<b>BSW Type</b>
J1939DCM_GREATER_THAN		EcucEnumerationLiteralDef
<b>BSW Description</b>		
<b>Template Description</b>		
<p><b>DiagnosticEnvConditionFormula.op:</b> This attribute represents the concrete operator (supported operators: and, or) of the condition formula.</p> <p><b>DiagnosticCompareTypeEnum.isGreaterThan:</b> greater than</p>		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isGreaterThan		
<b>Mapping Rule</b>		<b>Mapping Type</b>
1:1 mapping		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00267

BSW Module	BSW Context	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType	
BSW Parameter	BSW Type	
J1939DCM_LESS_OR_EQUAL	EcucEnumerationLiteralDef	
BSW Description		
Template Description		
<p><b>DiagnosticEnvConditionFormula.op:</b> This attribute represents the concrete operator (supported operators: and, or) of the condition formula.</p> <p><b>DiagnosticCompareTypeEnum.isLessOrEqual:</b> less than or equal</p>		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isLessOrEqual		
Mapping Rule		Mapping Type
		full
Mapping Status		Mapping ID
valid		up_Dcm_00270

BSW Module	BSW Context	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType	
BSW Parameter	BSW Type	
J1939DCM_LESS_THAN	EcucEnumerationLiteralDef	
BSW Description		
Template Description		
<p><b>DiagnosticEnvConditionFormula.op:</b> This attribute represents the concrete operator (supported operators: and, or) of the condition formula.</p> <p><b>DiagnosticCompareTypeEnum.isLessThan:</b> less than</p>		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isLessThan		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00268

BSW Module	BSW Context	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition	
BSW Parameter	BSW Type	



J1939DcmSwcModeRef	EcucInstanceReferenceDef
<b>BSW Description</b>	
This parameter references a mode in a particular mode request port of a software component that is used for the condition.	
<b>Template Description</b>	
This reference represents both the ModeDeclarationGroupPrototype and the ModeDeclaration relevant for the mode comparison.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvModeCondition.modeElement	
<b>Mapping Rule</b>	<b>Mapping Type</b>
For DcmModeRef a new DiagnosticEnvSwcModeElement is used, pointing to the ModeDeclaration via PModeInSystemInstanceRef. This new DiagnosticEnvModeElement shall be aggregated by the same DiagnosticEnvironmentalCondition as the DiagnosticEnvModeCondition, in which the target of the reference DiagnosticEnvModeCondition.modeElement points to the this DiagnosticEnvModeElement.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00272

<b>BSW Module</b>	<b>BSW Context</b>	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition	
<b>BSW Parameter</b>	<b>BSW Type</b>	
J1939DcmSwcSRDataElementRef	EcucReferenceDef	
<b>BSW Description</b>		
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.		
<b>Template Description</b>		
This reference represents the related diagnostic data element.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvDataCondition.dataElement		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
1:1 mapping	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid	up_Dcm_00274	

<b>BSW Module</b>	<b>BSW Context</b>	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmSwcSRDataElementValue/J1939DcmSwcSRDataElementArray/J1939DcmSwcSRDataElementArrayElement	
<b>BSW Parameter</b>	<b>BSW Type</b>	
J1939DcmSwcSRDataElementArrayElementIndex	EcucIntegerParamDef	
<b>BSW Description</b>		
Index to an array SR data element.		
<b>Template Description</b>		
This attribute represents a fixed compare value taken to evaluate the compare condition.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvDataCondition.compareValue		

<b>Mapping Rule</b>	<b>Mapping Type</b>
DiagnosticDataElement referenced by the DiagnosticEnvDataCondition is an array.	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00298

<b>BSW Module</b>	<b>BSW Context</b>	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmSwcSRDataElementValue/J1939DcmSwcSRDataElementArray/J1939DcmSwcSRDataElementArrayElement	
<b>BSW Parameter</b>		<b>BSW Type</b>
J1939DcmSwcSRDataElementArrayElementValue		EcucIntegerParamDef
<b>BSW Description</b>		
Value of an array SR data element compare value.		
<b>Template Description</b>		
This attribute represents a fixed compare value taken to evaluate the compare condition.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvDataCondition.compareValue		
<b>Mapping Rule</b>		<b>Mapping Type</b>
DiagnosticDataElement referenced by the DiagnosticEnvDataCondition is an array.		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00299

<b>BSW Module</b>	<b>BSW Context</b>	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmSwcSRDataElementValue/J1939DcmSwcSRDataElementPrimitive	
<b>BSW Parameter</b>		<b>BSW Type</b>
J1939DcmSwcSRDataElementPrimitiveValue		EcucIntegerParamDef
<b>BSW Description</b>		
Reference to a primitive SR data element compare value.		
<b>Template Description</b>		
This attribute represents a fixed compare value taken to evaluate the compare condition.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvDataCondition.compareValue		
<b>Mapping Rule</b>		<b>Mapping Type</b>
DiagnosticDataElement referenced by the DiagnosticEnvDataCondition is primitive.		full
<b>Mapping Status</b>		<b>Mapping ID</b>
valid		up_Dcm_00297

<b>BSW Module</b>	<b>BSW Context</b>	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions	
<b>BSW Parameter</b>		<b>BSW Type</b>
J1939DcmModeRule		EcucParamConfContainerDef
<b>BSW Description</b>		

<p>This container contains the configuration of a mode rule which represents a logical expression with J1939DcmModeConditions or other J1939DcmModeRules as arguments.</p> <p>All arguments are processed with the operator defined by DcmLogicalOperator, for instance: Argument_A AND Argument_B AND Argument_C</p>	
<b>Template Description</b>	
<p>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.</p> <p>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.</p>	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula	
<b>Mapping Rule</b>	<b>Mapping Type</b>
<p>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 DiagnosticEnvironmentalCondition shall be created with DiagnosticEnvironmentalCondition.formula containing a DiagnosticEnvConditionFormula. In both cases, if no DcmLogicalOperator is present in this DcmModeRule, then DiagnosticEnvConditionFormula shall be set to DiagnosticLogicalOperatorEnum.logicalAnd.</p>	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	

<b>BSW Module</b>	<b>BSW Context</b>	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeRule	
<b>BSW Parameter</b>		<b>BSW Type</b>
J1939DcmArgumentRef		EcucChoiceReferenceDef
<b>BSW Description</b>		
This is a choice reference either to a mode condition or a an other mode rule serving as sub-expression.		
<b>Template Description</b>		
A DiagnosticEnvConditionFormulaPart can either be a atomic condition, e.g. a DiagnosticEnvCompareCondition, or a DiagnosticEnvConditionFormula, again, which allows arbitrary nesting.		
<b>M2 Parameter</b>		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormulaPart		
<b>Mapping Rule</b>	<b>Mapping Type</b>	
<p>Depending on the destination, one DcmArgumentRef is mapped to a DiagnosticEnvConditionFormula if "destination" is a DcmModeRule, and to a DiagnosticEnvCompareCondition, if "destination" is a DcmModeCondition. The order of the aggregation of the DiagnosticEnvConditionFormulaParts shall correspond to the ordering of the index of the DcmArgumentRefs.</p>	full	
<b>Mapping Status</b>	<b>Mapping ID</b>	
valid	up_Dcm_00264	

BSW Module	BSW Context	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeRule/J1939DcmLogicalOperator	
BSW Parameter		BSW Type
J1939DCM_AND		EcucEnumerationLiteralDef
BSW Description		
Template Description		
<b>DiagnosticEnvConditionFormula.op:</b> This attribute represents the concrete operator (supported operators: and, or) of the condition formula.		
<b>DiagnosticLogicalOperatorEnum.logicalAnd:</b> Logical AND		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticLogicalOperatorEnum.logicalAnd		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00261

BSW Module	BSW Context	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeRule/J1939DcmLogicalOperator	
BSW Parameter		BSW Type
J1939DCM_OR		EcucEnumerationLiteralDef
BSW Description		
Template Description		
<b>DiagnosticEnvConditionFormula.op:</b> This attribute represents the concrete operator (supported operators: and, or) of the condition formula.		
<b>DiagnosticLogicalOperatorEnum.logicalOr:</b> Logical OR		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticLogicalOperatorEnum.logicalOr		
Mapping Rule		Mapping Type
1:1 mapping		full
Mapping Status		Mapping ID
valid		up_Dcm_00262

BSW Module	BSW Context	
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeRule	
BSW Parameter		BSW Type

J1939DcmModeRuleNrcValue	EcucIntegerParamDef
<b>BSW Description</b>	
Optional parameter which defines the NRC to be sent in case the mode rule condition is not valid.	
<b>Template Description</b>	
This attribute represents the concrete NRC value that shall be returned if the condition fails.	
<b>M2 Parameter</b>	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvCondition Formula.nrcValue	
<b>Mapping Rule</b>	<b>Mapping Type</b>
1:1 mapping	full
<b>Mapping Status</b>	<b>Mapping ID</b>
valid	up_Dcm_00263

## F Splitable Elements in the Scope of this Document

This chapter contains a table of all model elements stereotyped `<<atpSplitable>>` in the scope of this document.

Each entry in Table F.1 consists of the identification of the specific model element itself and the applicable value of the tagged value `atp.Splitkey`.

For more information about the concept of splitable model elements and how these shall be treated please refer to [22].

Name of splitable element	Splitkey
<a href="#">DiagnosticAging.agingCycle</a>	agingCycle, variationPoint.ShortLabel
<a href="#">DiagnosticContributionSet.commonProperties</a>	commonProperties
<a href="#">DiagnosticContributionSet.element</a>	element, variationPoint.shortLabel
<a href="#">DiagnosticContributionSet.serviceTable</a>	serviceTable, variationPoint.shortLabel
<a href="#">DiagnosticDataIdentifier.dataElement</a>	dataElement, variationPoint.shortLabel
<a href="#">DiagnosticEcuInstanceProps.ecuInstance</a>	ecuInstance
<a href="#">DiagnosticEnableConditionGroup.enableCondition</a>	enableCondition, variationPoint.shortLabel
<a href="#">DiagnosticEvent.connectedIndicator</a>	shortName, variationPoint.shortLabel
<a href="#">DiagnosticInfoType.dataElement</a>	dataElement
<a href="#">DiagnosticParameter.dataElement</a>	shortName, variationPoint.shortLabel
<a href="#">DiagnosticParameterIdentifier.dataElement</a>	dataElement, variationPoint.shortLabel
<a href="#">DiagnosticProtocol.diagnosticConnection</a>	diagnosticConnection, variationPoint.shortLabel
<a href="#">DiagnosticProtocol.serviceTable</a>	serviceTable, variationPoint.shortLabel
<a href="#">DiagnosticSecurityAccess.securityLevel</a>	securityLevel
<a href="#">DiagnosticServiceTable.diagnosticConnection</a>	diagnosticConnection, variationPoint.shortLabel
<a href="#">DiagnosticStorageConditionGroup.storageCondition</a>	storageCondition, variationPoint.shortLabel
<a href="#">DiagnosticTroubleCodeGroup.dtc</a>	dtc, variationPoint.shortLabel
<a href="#">DiagnosticTroubleCodeProps.extendedDataRecord</a>	shortName, variationPoint.shortLabel
<a href="#">DiagnosticTroubleCodeProps.freezeFrame</a>	shortName, variationPoint.shortLabel

**Table F.1: Usage of splitable elements**

## G 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 Table G.1 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 [22].

Variation Point	Latest Binding Time
<a href="#">DiagnosticAging.agingCycle</a>	preCompileTime
<a href="#">DiagnosticAging.threshold</a>	preCompileTime
<a href="#">DiagnosticCommonProps</a>	codeGenerationTime
<a href="#">DiagnosticContributionSet.element</a>	postBuild
<a href="#">DiagnosticContributionSet.serviceTable</a>	postBuild
<a href="#">DiagnosticDataIdentifier.dataElement</a>	postBuild
<a href="#">DiagnosticEnableConditionGroup.enableCondition</a>	postBuild
<a href="#">DiagnosticEvent.connectedIndicator</a>	postBuild
<a href="#">DiagnosticEvent.eventFailureCycleCounterThreshold</a>	postBuild
<a href="#">DiagnosticFreezeFrame.recordNumber</a>	preCompileTime
<a href="#">DiagnosticIndicator.healingCycleCounterThreshold</a>	preCompileTime
<a href="#">DiagnosticParameter.dataElement</a>	postBuild
<a href="#">DiagnosticParameterIdentifier.dataElement</a>	postBuild
<a href="#">DiagnosticProtocol.diagnosticConnection</a>	postBuild
<a href="#">DiagnosticProtocol.serviceTable</a>	postBuild
<a href="#">DiagnosticRoutine.id</a>	preCompileTime
<a href="#">DiagnosticServiceTable.diagnosticConnection</a>	postBuild
<a href="#">DiagnosticStorageConditionGroup.storageCondition</a>	postBuild
<a href="#">DiagnosticTroubleCodeGroup.dtc</a>	postBuild
<a href="#">DiagnosticTroubleCodeGroup.groupNumber</a>	preCompileTime
<a href="#">DiagnosticTroubleCodeObd.considerPtoStatus</a>	preCompileTime
<a href="#">DiagnosticTroubleCodeObd.obdDTCValue</a>	preCompileTime
<a href="#">DiagnosticTroubleCodeProps.extendedDataRecord</a>	preCompileTime
<a href="#">DiagnosticTroubleCodeProps.freezeFrame</a>	preCompileTime

**Table G.1: Usage of variation points**