



DRAFT INTERNATIONAL STANDARD ISO/DIS 22901-2

ISO/TC 22/SC 3

Secretariat: DIN

Voting begins on:
2009-07-09

Voting terminates on:
2009-12-09

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION

Road vehicles — Open diagnostic data exchange (ODX) — Part 2: Emissions-related diagnostic data

*Véhicules routiers — Échange de données de diagnostic ouvert (ODX) —
Partie 2: Données de diagnostic relatives aux émissions*

ICS 43.180

In accordance with the provisions of Council Resolution 15/1993 this document is circulated in the English language only.

Conformément aux dispositions de la Résolution du Conseil 15/1993, ce document est distribué en version anglaise seulement.

To expedite distribution, this document is circulated as received from the committee secretariat. ISO Central Secretariat work of editing and text composition will be undertaken at publication stage.

Pour accélérer la distribution, le présent document est distribué tel qu'il est parvenu du secrétariat du comité. Le travail de rédaction et de composition de texte sera effectué au Secrétariat central de l'ISO au stade de publication.

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENT AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

Copyright notice

This ISO document is a Draft International Standard and is copyright-protected by ISO. Except as permitted under the applicable laws of the user's country, neither this ISO draft nor any extract from it may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission being secured.

Requests for permission to reproduce should be addressed to either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Reproduction may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

Copyright notice

This ISO document is a Draft International Standard and is copyright-protected by ISO. Except as permitted under the applicable laws of the user's country, neither this ISO draft nor any extract from it may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission being secured.

Requests for permission to reproduce should be addressed to either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Reproduction may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

Contents

Page

Foreword.....	vi
Introduction	vii
1 Scope	1
2 Normative references	2
3 Terms, definitions, symbols and abbreviated terms.....	3
3.1 Terms and definitions	3
3.2 Symbols and abbreviated terms	3
3.3 Conventions	3
4 Emissions-related OBD ODX use cases.....	4
4.1 Use case 1 — OBD Scan Tool based on a Modular VCI architecture and ODX	4
4.2 Use case 2 — Conversion of emissions-related OBD data to ODX format.....	5
5 Emissions-related OBD ODX application examples	7
5.1 OBD conformance tester according to SAE J1699-3	7
5.2 Usage of ODX as a configuration for standardized ECU software	9
5.3 Usage of ODX checker rules for ECU development.....	10
6 Specification release version information	10
6.1 Specification release version location	10
6.2 Specification release version	10
7 OBD authoring in ODX	11
7.1 ODX layering	11
7.1.1 Relationship between ODX layers.....	11
7.1.2 Authoring of Functional Groups	12
7.1.3 Authoring of emissions-related protocols	12
7.1.4 Authoring of emissions-related ECU-SHARED-DATA	13
7.1.5 Authoring of VEHICLE-INFO.....	13
7.2 Service implementation in ODX	14
7.2.1 General.....	14
7.2.2 OBD Services authoring	14
7.2.3 ODX REQUEST implementation.....	17
7.2.4 ODX POS-RESPONSE implementation	17
7.3 ODX PARAMs implementation	18
7.3.1 Service IDs (SID)	19
7.3.2 Local Identifier implementation.....	20
7.3.3 Service-specific parameters	21
7.4 Conversion of PIDs to ODX	25
7.4.1 Properties of PIDs.....	25
7.4.2 Rules for type “Linear”	26
7.4.3 Rules for type “Boolean”	26
7.4.4 Rules for type “BitSelect”	27
7.4.5 Rules for type “BitSet”	27
7.4.6 Rules for type “Number2Text”	28
7.4.7 Rules for type “Number”	28
7.4.8 Rules for type “DTC”	29
7.4.9 Special handling of PID 0x13 and 0x1D.....	29
7.5 Conversion of DTCs to ODX.....	31
7.6 ODX samples of ISO°15031-5 services and authored data	33
7.6.1 General.....	33
7.6.2 Service 0x01 — Request current powertrain diagnostic data.....	33

7.6.3	Service 0x02 — Request powertrain freeze frame data.....	39
7.6.4	Service 0x03, 0x07 and 0x0A — Request emissions-related DTC	42
7.6.5	Service 0x04 — Clear/Reset emission-related diagnostic information	45
7.6.6	Service 0x06 — Request on-board monitoring test results	47
7.6.7	Service 0x08 — Request control of on-board device	52
7.6.8	Service 0x09 — Request vehicle information.....	54
7.6.9	Authored data — PIDs.....	57
7.6.10	Authored data — DTCs	68

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 22901-2 was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 3, *Electrical and electronic equipment*.

ISO 22901 consists of the following parts, under the general title *Road vehicles — Open diagnostic data exchange (ODX)*:

- *Part 1: Data model specification*
- *Part 2: Emissions-related diagnostic data*

Introduction

This International Standard has been established in order to define the data format for transferring standardized emissions-related diagnostic data of the vehicle's OBD system between system supplier, vehicle manufacturer and service dealerships and diagnostic tools of different vendors.

The standardized information is contained in the following standards:

- Diagnostic protocol information:
 - ISO 15765-4, Road vehicles — Diagnostics on Controller Area Network (CAN) —Part 4: Requirements for emissions-related systems,
 - ISO 14230-4:2000, Road vehicles — Keyword protocol 2000 for diagnostic systems—Part 4: Requirements for emissions-related systems,
 - ISO 9141-2: 1994, Road vehicles — Diagnostic systems—Part 2: CARB requirements for interchange of digital information,
 - ISO 9141-2: 1994/Amd.1:1996, Road vehicles—Diagnostic systems—Part 2: CARB requirements for interchange of digital information Amendment 1,
 - SAE°J1850: ISO°15031-5 Emissions-related diagnostic services,
- Emissions-related OBD data:
 - ISO 15031-2, Road vehicles — Communication between vehicle and external test equipment for emissions-related diagnostics — Part 2: Terms, definitions, abbreviations, and acronyms,
 - SAE J1930-DA, Digital Annex, Electrical/Electronic Systems Diagnostic Terms, Definitions, Abbreviations, and Acronyms,
 - ISO 15031-5, Road vehicles — Communication between vehicle and external equipment for emissionsrelated diagnostics — Part 5: Emissions-related diagnostic services,
 - SAE J1979-DA, Digital Annex, E/E Diagnostic Test Modes,
 - ISO 15031-6, Road vehicles—Communication between vehicle and external test equipment for emissions-related diagnostics—Part 6: Diagnostic Trouble Code Definitions,
 - SAE J2012-DA, Digital Annex, Diagnostic Trouble Code Definitions,
 - ISO 15031-4, Road vehicles — Communication between vehicle and external equipment for emissionsrelated diagnostics — Part 4: External test equipment;
- OBD Conformance test cases:
 - SAE J1699-3, OBD II Compliance Test Cases;

The automotive industry mostly utilizes an informal description to document diagnostic data stream information of vehicle ECUs. Each user, who desires to use the ECU diagnostic data stream documentation to setup development tools or service diagnostic test equipment, needs a manual transformation of this

documentation into a format readable by these tools. This effort will no longer be required if the diagnostic data stream information is provided in ODX format and if those tools support the ODX format.

Road vehicles — Open diagnostic data exchange (ODX) — Part 2: Emissions-related diagnostic data

1 Scope

This part of Road vehicles – Open diagnostic data exchange (ODX) - specifies the concept of utilizing a new industry standard diagnostic format to make diagnostic data stream information available to diagnostic tool application manufacturer to simplify the support of the aftermarket automotive service industry. The ODX modelled diagnostic data are compatible to the software requirements of the Modular Vehicle Communication Interface (ISO 22900-2 and -3). The ODX modelled diagnostic data will enable a MVDI device to communicate with the vehicle (ECU(s)) and interpret the diagnostic data contained in the messages exchanged between the external test equipment and the ECU(s). For an ODX compliant external test equipment no software programming is necessary to convert diagnostic data into technician readable information to be displayed by the tester.

The present specification contains emissions-related OBD data examples described in ODX. The data examples derive from the ISO 15031 (all parts) standard. Examples are: diagnostic trouble codes, data parameters, identification data and communication parameters.

The emissions-related OBD ODX modelled diagnostic data describe

- Protocol specification from diagnostic communication of emissions-related ECUs,
- Communication parameters for the emissions-related OBD protocols and data link layers and for emissions-related ECU software,
- Related vehicle interface description (connectors and pin-out),
- Functional description of diagnostic capabilities of a network of ECUs;

Figure 1 — Usage of ODX data in the ECU life cycle” shows the usage of ODX in the ECU life cycle. Engineering, manufacturing, and service specify communication protocol and data to be implemented in the ECU. This information will be documented in a structured format utilizing the XML standard and by an appropriate ODX authoring tool. There is potential to generate ECU software from the ODX file. Furthermore, the same ODX file is used to setup the diagnostic engineering tools to verify proper communication with the ECU and to perform functional verification and compliance testing. Once all quality goals are met, the ODX file may be released to a diagnostic database. Diagnostic information is now available to manufacturing, service, OEM franchised dealers, and aftermarket service outlets via Intranet and Internet.

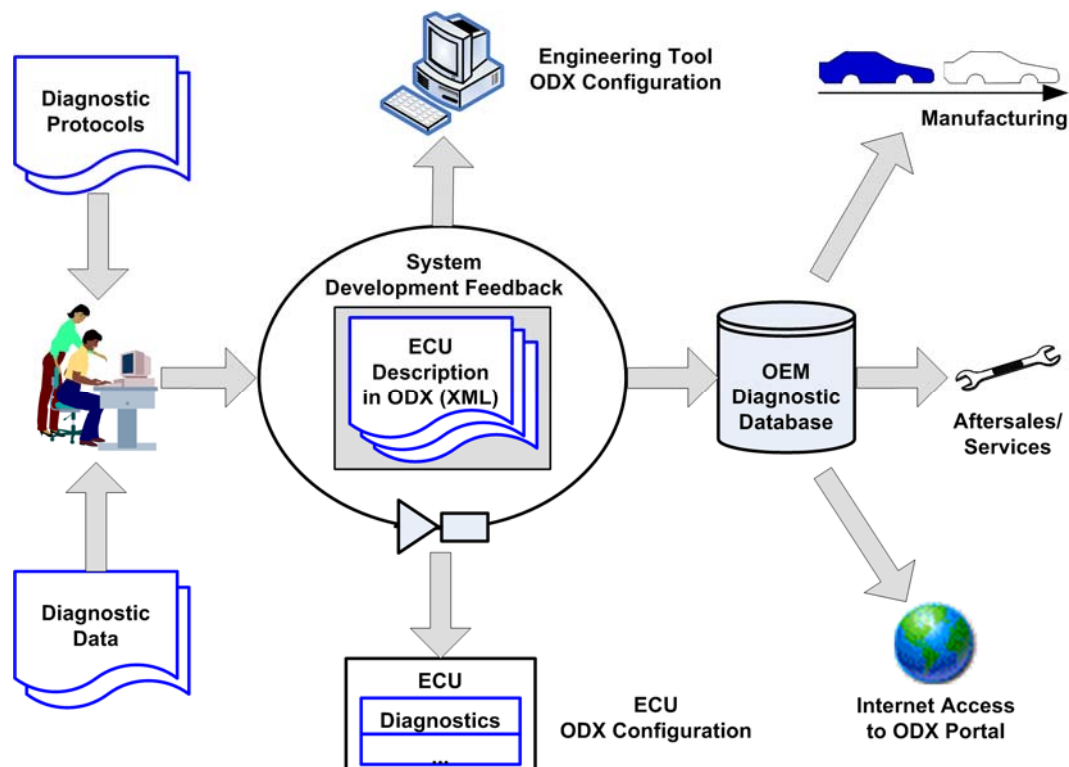


Figure 1 — Usage of ODX data in the ECU life cycle

The objective of this specification is to ensure that diagnostic data from any vehicle manufacturer is independent of the testing hardware and protocol software supplied by any test equipment manufacturer.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 7498-1, *Information processing systems - Open systems interconnection - Basic reference model*

ISO/IEC 10731, *Information technology -- Open Systems Interconnection -- Basic Reference Model -- Conventions for the definition of OSI services*

ISO 9141-2: 1994, *Road vehicles —Diagnostic systems —Part 2: CARB requirements for interchange of digital information*

ISO 9141-2: 1994/Amd.1:1996, *Road vehicles —Diagnostic systems —Part 2: CARB requirements for interchange of digital information Amendment 1*

ISO 14230-4, *Road vehicles — Keyword protocol 2000 for diagnostic systems —Part 4: Requirements for emissions-related systems*

ISO 15765-4, *Road vehicles — Diagnostics on Controller Area Network (CAN) —Part 4: Requirements for emissions-related systems*

ISO 15031 (all parts), *Road vehicles — Communication between vehicle and external test equipment for emissions-related diagnostics*

ISO 22900-2, *Road vehicles — Modular vehicle communication interfaces (MVCI) — Part 2: Diagnostic protocol data unit application programmer interface (D-PDU API)*

ISO 22900-3, *Road vehicles — Modular vehicle communication interfaces (MVCI) — Part 3: Diagnostic server application programming interface (D-server API)*

SAE J1699-3, *OBD II Compliance Test Cases*

SAE J1930-DA, *Digital Annex, Electrical/Electronic Systems Diagnostic Terms, Definitions, Abbreviations, and Acronyms*

SAE J1979-DA, *Digital Annex, Emissions-related Diagnostic Data Definition*

SAE J2012-DA, *Digital Annex, Diagnostic Trouble Code Definition*

3 Terms, definitions, symbols and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 22901-1 apply.

3.2 Symbols and abbreviated terms

MVCI Modular Vehicle Communication Interface

ODX-RT Open Diagnostic data eXchange – Run-Time format

3.3 Conventions

ISO 22901-2 is based on the conventions discussed in the O.S.I. Service Conventions (ISO/IEC 10731:1994) as they apply for diagnostic services.

4 Emissions-related OBD ODX use cases

4.1 Use case 1 — OBD Scan Tool based on a Modular VCI architecture and ODX

This use case describes the usage of an OBD scan tool compliant to ISO 15031-4 / SAE J1978 and implemented according to the Modular VCI specification (see ISO 22900-1, -2 and -3) and ODX (see ISO 22901-1).

The benefits of an emissions-related OBD scan tool which is based on the Modular VCI and ODX standard are the following:

- No software programming to support the implementation of:
 - new diagnostic trouble codes (see ISO 15031-6 / SAE J2012-DA),
 - new PIDs, Test IDs, Monitor Ids, INFOTYPES and Scaling Ids (see ISO°15031-5 / SAE°J1979-DA),
 - new standardized Terms, abbreviations and definitions (see ISO 15031-2 / SAE J1930-DA);
- OBD scan tool applications according to ISO 15031-4 are developed only once and are not impacted by modifications / changes in the definition of emissions-related OBD data and formats,
- Separation of application, communication logic and data items,

NOTE The Modular VCI software architecture supports the emissions-related OBD scan tool requirements as well as enhanced diagnostic protocols, data streams and applications.

Figure 2 — OBD scan tool based on Modular VCI architecture and ODX illustrates an external test equipment connected to the vehicle's diagnostic connector. The OBD scan tool's software architecture is compliant to the Modular VCI specifications. The diagnostic kernel is the key software component of the Modular VCI system. It implements the D-PDU API (see ISO 22900-2), the D-Server API (see ISO 22900-3) and the interface to the ODX derived runtime data.

The OBD scan tool application depends on standardized names or naming conventions as defined by this part of ISO 22901. These names are defined in the emissions-related ODX data and utilized by the OBD scan tool application to address logical links, services, and emission-related data. Using the standardized names and structures from this standard the interface to implement the scan tool application against is clearly defined. This is indicated by the dashed line in the figure.

The D-PDU API is a software component of the tool supplier's Modular VCI protocol module. It connects the diagnostic kernel with any Modular VCI compatible vehicle communication interface.

The D-Server API of the diagnostic kernel provides a standardized interface to the OBD scan tool applications. These applications shall be according to ISO 15031-4 which implements the standardized data and messages of ISO°15031-5 and -6.

The emissions-related ODX runtime data format is tool supplier specific. The runtime format is not contained in the ODX standard (see ISO 22901-1). Based on the use cases supported by the diagnostic tool the content and structure of the ODX runtime data format and content may differ. However, for emissions-related OBD the OBD scan tool applications and ODX runtime data shall support the full scope of ISO 15031 and the respective SAE J documents.

All emissions-related OBD data as specified in ISO 15031-2 and SAE J1930-DA, ISO°15031-5 and SAE°J1979-DA, ISO 15031-6 and SAE J2012-DA, SAE J1939 shall be authored according to the requirements established in this part of ISO 22901.

This use case requires the unique and complete definition of all elements necessary for any OBD scan tool application compliant to ISO 22900.

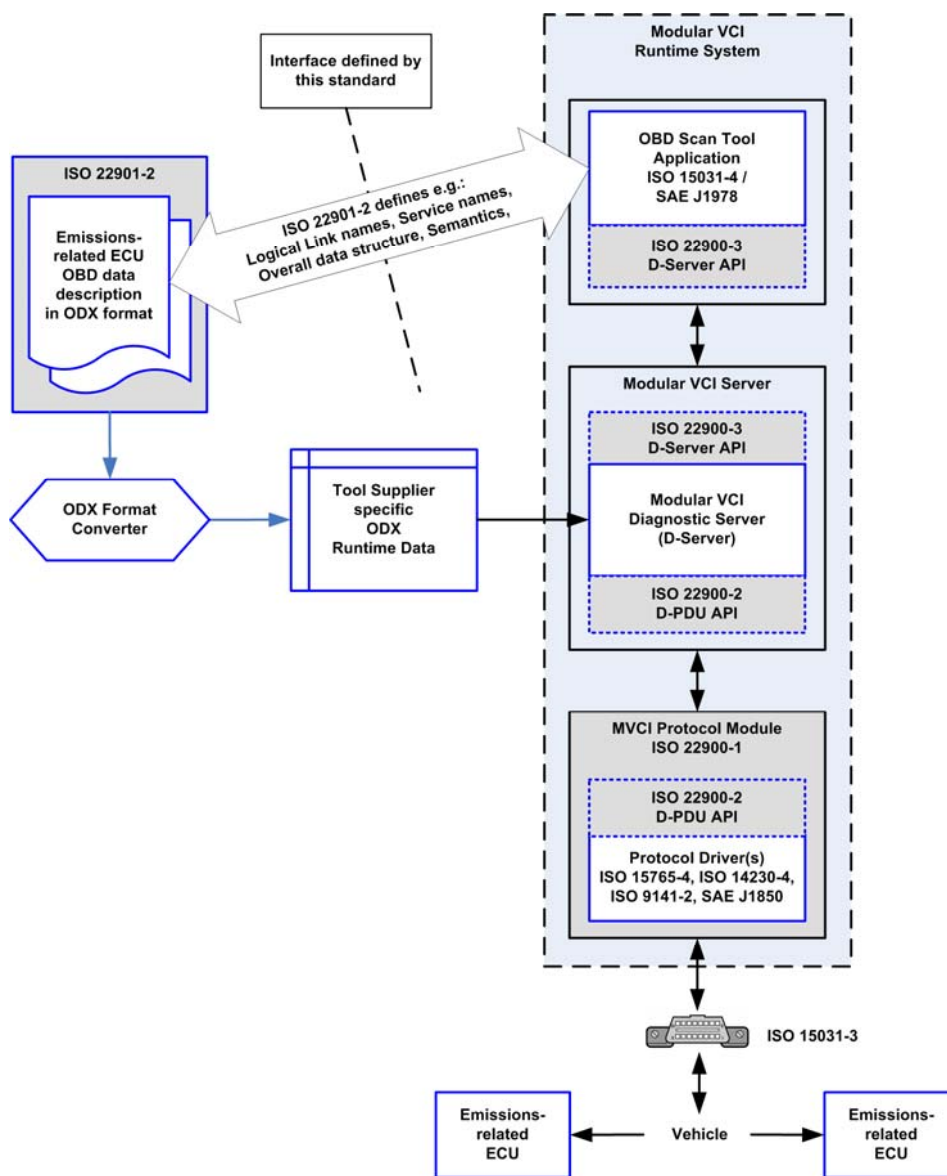


Figure 2 — OBD scan tool based on Modular VCI architecture and ODX

4.2 Use case 2 — Conversion of emissions-related OBD data to ODX format

This use case describes the conversion of emissions-related OBD data into the ODX format in order to provide various applications of external test equipment with emissions-related OBD data in an ODX-RT (runtime) format.

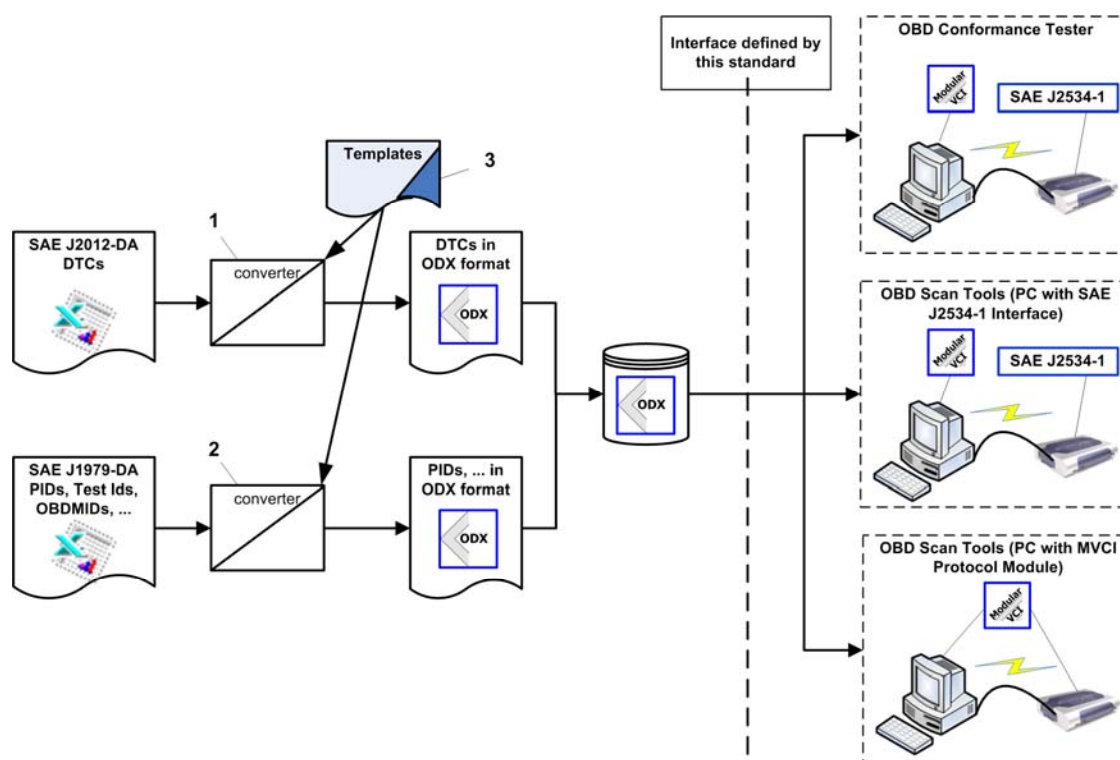
It is assumed that the external test equipment is based on the ISO 22900 Road vehicles – Modular VCI standard architecture.

The emissions-related OBD data files derive from the Registration Authority installation for ISO 15031.

The applicable emissions-related OBD data files are:

- SAE J1930-DA, Digital Annex, Electrical/Electronic Systems Diagnostic Terms, Definitions, Abbreviations, and Acronyms,
- SAE J1979-DA, Digital Annex, Emissions-related Diagnostic Data Definition,
- SAE J2012-DA, Digital Annex, Diagnostic Trouble Code Definition;

Figure 3 — Emissions-related OBD data converter to ODX-RT format illustrates the process to be followed in order to convert SAE J2012-DA, and SAE J1979-DA data file information (i.e. Excel or equivalent format) into a standardized ODX format [1, 2] which enrich the emissions-related OBD data with template [3] information.



Key

- 1 SAE J2012-DA DTC converter into ODX format
- 2 SAE J1979-DA PIDs, OBD MIDs, ... converter into ODX format
- 3 ODX converter templates to determine standardized ODX parts i.e. PROTOCOLS, COMPARAMS, ... and ODX usage as defined by this standard

Figure 3 — Emissions-related OBD data converter to ODX-RT format

The benefits of implementing this use case are:

- The setup/update of an ISO 22900 Modular VCI based OBD test equipment utilizing a SAE J2534-1 compliant vehicle communication interface or an MVCI compliant Protocol Module with emissions-related OBD data (ODX-RT format) which derive from a conversion of the applicable SAE Digital Annexes,

- The setup/update of an ISO 22900 Modular VCI based OBD conformance tester with emissions-related OBD data (ODX-RT format) which implements the test cases as specified in SAE J1699-3.

5 Emissions-related OBD ODX application examples

5.1 OBD conformance tester according to SAE J1699-3

This application example describes the implementation of an OBD conformance tester compliant to SAE J1699-3 and based on the Modular VCI software architecture. The base architecture as shown in use case 1 applies. The major difference between the emissions-related OBD scan tool and the OBD conformance tester is implemented in the test applications. While the emissions-related OBD scan tool is compliant to ISO 15031-4, the OBD conformance tester is compliant to SAE J1699-3. This specification describes very specific test cases in order to achieve vehicle emissions-related system compliance. These test cases have been introduced and referenced by legislation in order to reduce emissions-related diagnostic software implementation deviations in the ECUs from ISO 15031 and the respective SAE J documents.

The benefits of an OBD conformance tester based on the Modular VCI and ODX standard are:

- No software programming to support the implementation of:
 - new diagnostic trouble codes (see ISO 15031-6 / SAE J2012-DA),
 - new PIDs, Test IDs, Monitor Ids, INFOTYPES and Scaling Ids (see ISO 15031-5 / SAE J1979-DA),
 - new standardized terms, abbreviations and definitions (see ISO 15031-2 / SAE J1930-DA);
- Conformance test applications implement the test logic but not the data items (derive from emissions-related ODX runtime),
- Clear separation of application and communication logic as well as from all data items,

Figure 4 — ODX emissions OBD Modular VCI based OBD conformance tester is based on the architecture as shown in use case 1.

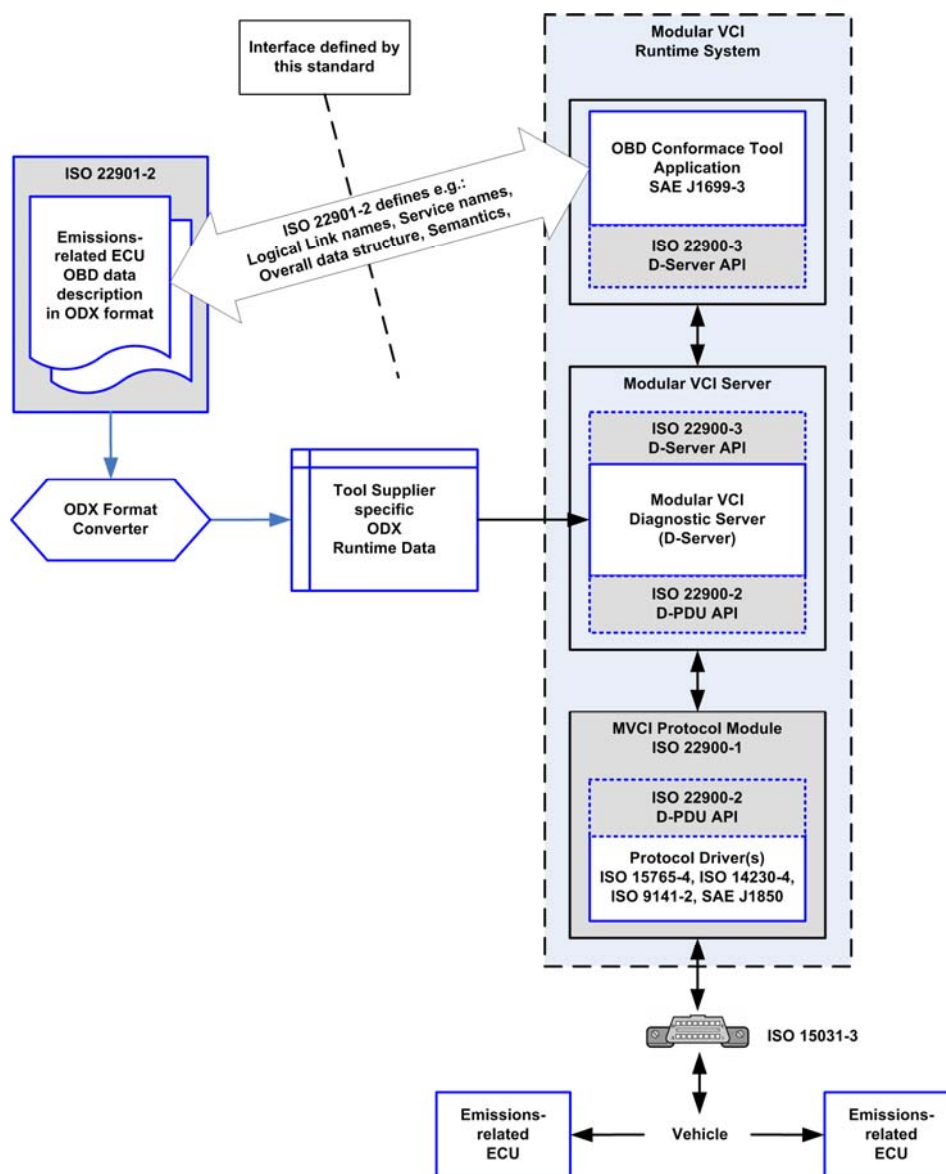


Figure 4 — ODX emissions OBD Modular VCI based OBD conformance tester

5.2 Usage of ODX as a configuration for standardized ECU software

This application example describes how to drive the implementation of the emissions-related OBD diagnostic software module of the ECU by the OBD ODX data. This can be done either by using the OBD ODX data as configuration for a generic diagnostic software module or by utilizing a software generation process which is controlled by the OBD ODX configuration data.

Once the OBD behaviour of an ECU is defined in ODX format, this file can be used to configure a standardized software part in the ECU.

The benefits of implementing this use case are:

- Standardized ECU software modules need to be tested only once,
- Standardized ECU software modules can be reused in different projects,
- The ECU behaviour fits exactly to the behaviour described in the ODX file (because the software as well as the documentation are derived from the same data source);

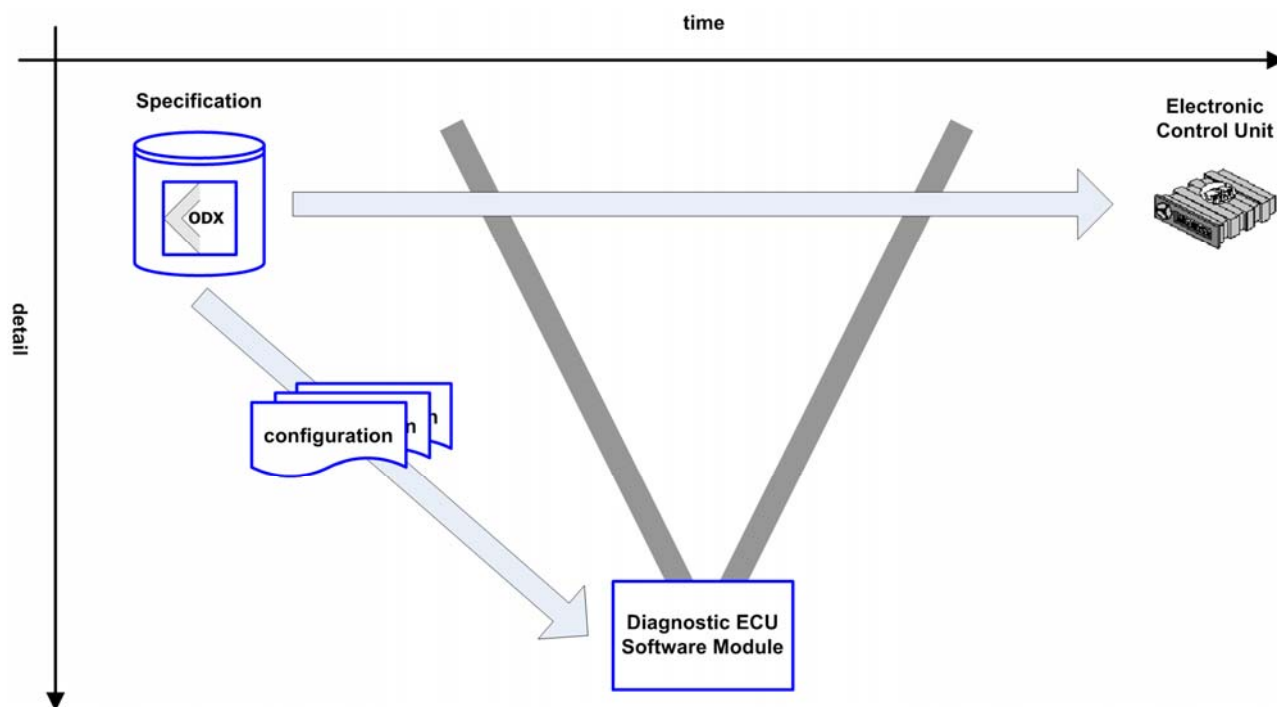


Figure 5 — Example of an ECU diagnostic software module and configuration data derived from ODX

5.3 Usage of ODX checker rules for ECU development

This application example describes the usage of ODX checker rules, which represent a subset of the SAE J1699-3 test cases.

For ODX adaptable checkers exist. These allow to check for ODX compliance and can be extended with individual checker rules. With these, OBD compliance can be checked before the ECU is implemented, only if the emissions-related OBD ODX data follow the requirements of this part of ISO 22901.

EXAMPLE When specifying the behaviour of an individual ECU in ODX, the support of Infotype 0x0A (ECU-name) for model year 2010 and later can be checked before the ECU code is implemented.

The benefits of implementing this application example are:

- Early check for errors (before ECU is implemented in the vehicle),
- Checker rules can be provided by a third party and made available to interested users;

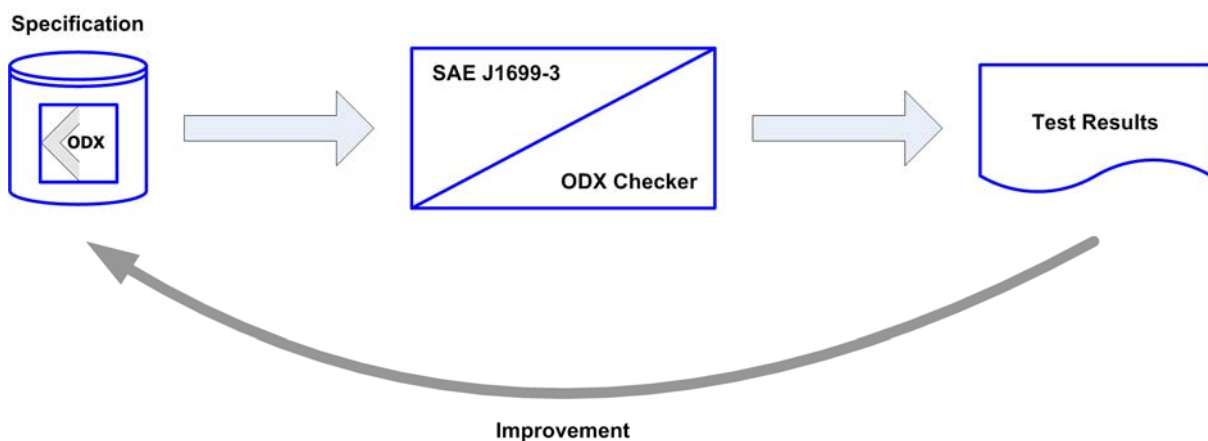


Figure 6 — Emissions-related OBD compliance test during ECU specification phase

6 Specification release version information

6.1 Specification release version location

The release version of the ODX standard can be obtained from every ODX file instance. It is contained in the MODEL-VERSION attribute.

```
<ODX MODEL-VERSION="2.2.0">
```

6.2 Specification release version

The specification release version of this document is: 2.2.0

7 OBD authoring in ODX

7.1 ODX layering

7.1.1 Relationship between ODX layers

Figure 7 — ComParam-Specs for emissions-related OBD protocols and data illustrates the partitioning of the emissions-related OBD protocols and their associated ComParamSpec from the ECU-Shared-Data and Functional Groups 1 and 2. The Vehicle-Info specifies the Logical Links to the Protocols and Functional Groups. The light and dotted parts are user extensions that might be integrated, if protocols other than ISO°15765-4 shall be supported.

This standard covers only the ISO°15765-4 case.

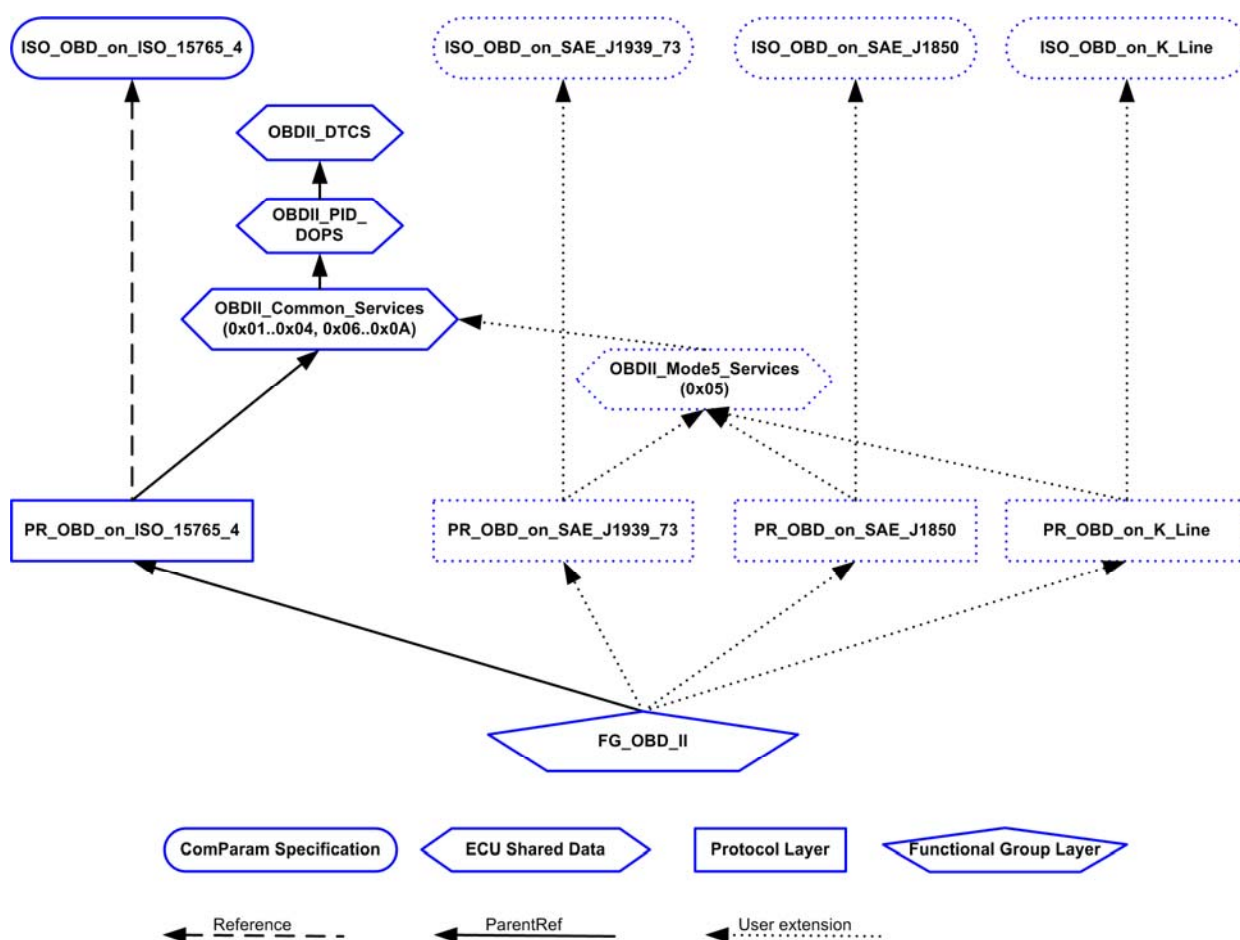


Figure 7 — ComParam-Specs for emissions-related OBD protocols and data

7.1.2 Authoring of Functional Groups

Functional Groups specify data for a group of emissions-related ECUs i.e. Engine Control Module and Transmission Control Module which contain all required information to enable the Modular VCI compliant emissions-related OBD test equipment to perform functional communication.

A Functional Group named “FG_OBD_II” specifies the data relevant to the OBD message protocol for all of the available and supported physical link layers (ISO_OBD_on_SAE_J1850, ISO_OBD_on_SAE_J1939_73,

ISO_OBD_on_K_Line, ISO_OBD_on_ISO_15765_4). The ComParamSpec as defined by ISO 22900-2 specify the protocol specific message framing, message timing and message addressing information.

7.1.3 Authoring of emissions-related protocols

The PROTOCOL class in ODX is used to capture communication data like message layout, parameters in diagnostic requests and responses, conversion information to convert from coded values to physical values and vice versa.

For emission-related data, four (4) ODX protocol layers named “PR_ISO_15031_5_on_ISO_15765_4”, “PR_ISO_15031_5_on_SAE_J1939_73”, “PR_ISO_15031_5_on_SAE_J1850” and “PR_ISO_15031_5_on_K_Line” are defined. They capture the physical layer and transport layer specific protocol information

Example Name tags of protocol PR_ISO_15031_5_on_ISO_15765_4

```
<PROTOCOL ID="ID_PR_ISO_15031_5_on_ISO_15765_4" TYPE="">
  <SHORT-NAME>PR_ISO_15031_5_on_ISO_15765_4</SHORT-NAME>
  <LONG-NAME>ISO OBD on CAN</LONG-NAME>
  <COMPARAM-SPEC-REF DOCTYPE="COMPARAM-SPEC" ID-
    REF="ID_ISO_OBD_on_ISO_15765_4"/>
</PROTOCOL>
```

Table 1 — Definition of SHORT-NAME and LONG-NAME of OBD protocols SHORT-NAME and LONG-NAME of the protocols defined in Figure 7 — ComParam-Specs for emissions-related OBD protocols and data even those that are only available by user extension.

Table 1 — Definition of SHORT-NAME and LONG-NAME of OBD protocols

SHORT-NAME	LONG-NAME
PR_ISO_15031_5_on_ISO_15765_4	ISO OBD on CAN
PR_ISO_15031_5_on_SAE_J1939_73	ISO OBD on Truck and Bus CAN
PR_ISO_15031_5_on_SAE_J1850	ISO OBD on J1850 VPW and J1850 PWM
PR_ISO_15031_5_on_K_Line	ISO OBD on 9141-2 K-Line and KWP2000 K-Line

In order to identify and group OBD services effectively, all OBD Services are members of the Functional Class “OBD.PROTOCOL.OBDOnCAN.FUNCT-CLASS.emissionRelatedDiagnosticServices”.

EXAMPLE FUNCT-CLASS

```
<FUNCT-CLASS>
  <FUNCT-CLASS ID="OBD.PROTOCOL.OBDOnCAN.FUNCT-CLASS.emissionRelatedDiagnosticServices">
    <SHORT-NAME>ISO_15031_5</SHORT-NAME>
    <LONG-NAME>ISO 15031-5</LONG-NAME>
  </FUNCT-CLASS>
</FUNCT-CLASS>
```

7.1.4 Authoring of emissions-related ECU-SHARED-DATA

Several named ECU-SHARED-DATA ODX containers grouped in an inheritance hierarchy capture OBDII relevant services as well as parameter encoding and decoding information.

“OBDII_DOPS” hold the encoding and decoding description of response and request parameter as well as units and dimension specifications.

“OBDII_Common_Services” holds OBDII services definitions for modes 0x01-0x04, 0x06-0x0A.

“OBDII_Mode5_Service” holds OBDII service definitions for mode 0x05. Mode 0x05 service is separated out because it is not used for OBD on CAN.

EXAMPLE Ecu-Shared-Data

```
<ECU-SHARED-DATA>
  <SHORT-NAME>OBDII_DOPS</SHORT-NAME>
  <LONG-NAME>DTCS for the OBD protocol</LONG-NAME>
  <DIAG-DATA-Dictionary-SPEC>
    <DTC-DOPS>
      <DTC-DOP ID="DIAG-LAYER-CONTAINER.OBDTemplate.DTC-DOP.J2012DTC">
        <SHORT-NAME>J2012DTC</SHORT-NAME>
        <LONG-NAME>J2012DTC</LONG-NAME>
        <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
          <BIT-LENGTH>16</BIT-LENGTH>
        </DIAG-CODED-TYPE>
        <PHYSICAL-TYPE BASE-DATA-TYPE="A_UINT32" DISPLAY-RADIX="HEX"/>
        <COMPU-METHOD>
          <CATEGORY>IDENTICAL</CATEGORY>
        </COMPU-METHOD>
        <DTCS>
          <DTC ID="ID_OBD.DTC-DOP.J2012DTC.DTC.P0000">
            <SHORT-NAME>P0000</SHORT-NAME>
            <TROUBLE-CODE>0</TROUBLE-CODE>
            <TEXT TI="0">ISO/SAE reserved - Use Not Allowed</TEXT>
          </DTC>
          ...
        </DTCS>
      </DTC-DOP>
    </DTC-DOPS>
    ...
  </DIAG-DATA-Dictionary-SPEC>
</ECU-SHARED-DATA>
```

7.1.5 Authoring of VEHICLE-INFO

The VEHICLE-INFO-SPECIFICATION “VI_OBDII” specifies the logical links to be supported by the Modular VCI (see ISO 22900).

All OBDII logical links refer to Functional Group “FG_OBD_II”. However, for different physical layers, the respective predefined protocol of chapter 7.1.3 is referenced.

There shall be four Logical Links defined named “LL_OBD_on_ISO_15765_4”, “LL_OBD_on_SAE_J1939_73”, “LL_OBD_on_SAE_J1850” and “LL_OBD_on_K_Line”:

- a) LL_OBD_on_ISO_15765_4 references FG_OBD_II and PR_OBD_on_ISO_15765_4,
- b) LL_OBD_on_SAE_J1939_73 references FG_OBD_II and PR_OBD_on_SAE_J1939_73,
- c) LL_OBD_on_SAE_J1850 references FG_OBD_II and PR_OBD_on_SAE_J1850,

d) LL_OBD_on_K_Line references FG_OBD_II and PR_OBD_on_K_Line;

This technique will ensure, that selecting a logical link is sufficient to establish ECU communication using the correct protocol, communication parameters and services.

IMPORTANT — OBD communication initialization is specified in ISO 22900-2.

7.2 Service implementation in ODX

7.2.1 General

Part 1 of ISO 22901 provides at least two alternatives to author diagnostic data:

- a) Dedicated diagnostic service related data authoring i.e. the same data (PIDS, INFOTYPES, ...) are used by all emissions-related OBD protocols and authored for each protocol redundantly,
- b) Diagnostic service independent related data authoring also called "table based authoring" i.e. the same data (PIDS, INFOTYPES, ...) are used by all emissions-related OBD protocols but authored only once and referenced by each protocol;

IMPORTANT — Emissions-related OBD data as specified in ISO 15031-5, -6 / SAE°J1979 and SAE°J1979-DA (Digital Annex) shall be authored according to alternative b).

7.2.2 OBD Services authoring

7.2.2.1 ODX IDs of OBD services

The ID of the OBD services shall be generated by the following rule: <Layer-ID>.<type prefix>_SHORT-NAME

```
<DIAG-SERVICE
  ID="ES_OBDIICommonServices.DiagnServi_Service01RequestCurrentPowertrainDiagnosticDataP
  ID13">
  <SHORT-NAME>Service01RequestCurrentPowertrainDiagnosticDataPID13</SHORT-NAME>
  <LONG-NAME>Service 0x01 - Request current powertrain diagnostic data (PID 0x13)</LONG-
  NAME>
  ...
</DIAG-SERVICE>
```

7.2.2.2 ODX DESCription of OBD services

The text of ISO°15031-5 clause "Functional description" of the particular OBD service shall be used.

7.2.2.3 ODX LONG-NAMES of OBD services

The LONG-NAME of the DIAG-SERVICE shall use the complete wording of the service description in the headlines of ISO°15031-5. This document distinguishes between services for ISO°9141-2, ISO°14230-4, SAE°J1850 and services for ISO°15765-4, but the descriptions of the headlines are the same at the services in both cases, this circumstance has to be considered in the LONG-NAME or later on in the description of the services itself. REQUEST and POS-/NEG-RESPONSES have to be kept in mind in this case.

Table 2 — LONG-NAMES for OBD services regarding ISO°15765-4

Headline used in ISO°15031-5-ISO 15765-4	LONG-NAME
Service 0x01 – Request current powertrain diagnostic data	Service 0x01 – Request current powertrain diagnostic data
Service 0x02 – Request powertrain freeze frame data	Service 0x02 – Request powertrain freeze frame data
Service 0x03 – Request emission-related diagnostic trouble codes	Service 0x03 – Request emission-related diagnostic trouble codes
Service 0x04 – Clear/reset emission-related diagnostic information	Service 0x04 – Clear/reset emission-related diagnostic information
Service 0x05 – Request oxygen sensor monitoring test results	Service 0x05 – Request oxygen sensor monitoring test results
Service 0x06 – Request on-board monitoring test results for specific monitored systems	Service 0x06 – Request on-board monitoring test results for specific monitored systems
Service 0x07 – Request emission-related diagnostic trouble codes detected during current or last completed driving cycle	Service 0x07 – Request emission-related diagnostic trouble codes detected during current or last completed driving cycle
Service 0x08 – Request control of on-board system, test or component	Service 0x08 – Request control of on-board system, test or component
Service 0x09 – Request vehicle information	Service 0x09 – Request vehicle information
Service 0x0A – Request emissions-related diagnostic trouble codes with permanent status	Service 0x0A – Request emissions-related diagnostic trouble codes with permanent status

The following exceptions have to be considered. With service 0x01 the behaviour for PID 0x13 and PID 0x1D must be distinguished. Therefore two services for service 0x01 need to be defined. The LONG-NAME of both services are extended by the marker (PID 0x13) and (PID 0x1D) respectively.

Table 3 — LONG-NAMES for OBD service 0x01 distinguishing PID 0x13 and PID 0x1D

Headline used in ISO°15031-5 – ISO°15765-4	LONG-NAME
Service 0x01 – Request current powertrain diagnostic data	Service 0x01 – Request current powertrain diagnostic data (PID 0x13)
Service 0x01 – Request current powertrain diagnostic data	Service 0x01 – Request current powertrain diagnostic data (PID 0x1D)

7.2.2.4 ODX SHORT-NAMES of OBD services

The SHORT-NAME derives from the LONG-NAME according to rule "L4" in clause 7.4.2.

Table 4 — SHORT-NAMES for OBD services regarding ISO°15765-4

LONG-NAME	SHORT-NAME
Service 0x01 – Request current powertrain diagnostic data	Service01RequestCurrentPowertrainDiagnosticData
Service 0x02 – Request powertrain freeze frame data	Service02RequestPowertrainFreezeFrameData
Service 0x03 – Request emissions-related diagnostic trouble codes	Service03RequestEmissionRelatedDiagnosticTroubleCodes
Service 0x04 – Clear/reset emissions-related diagnostic information	Service04ClearResetEmissionRelatedDiagnosticInformation
Service 0x05 – Request oxygen sensor monitoring test results	Service05RequestOxygenSensorMonitoringTestResults
Service 0x06 – Request on-board monitoring test results for specific monitored systems	Service06RequestOnBoardMonitoringTestResults ForSpecificMonitoredSystems
Service 0x07 – Request emissions-related diagnostic trouble codes detected during current or last completed driving cycle	Service07RequestEmissionRelatedDiagnosticTroubleCodesDetectedDuringCurrent OrLastCompletedDrivingCycle
Service 0x08 – Request control of on-board system, test or component	Service08RequestControlOfOnBoardSystemTestOrComponent
Service 0x09 – Request vehicle information	Service09RequestVehicleInformation
Service 0x0A – Request emissions-related diagnostic trouble codes with permanent status	Service0ARequestEmissionRelatedDiagnosticTroubleCodesWithPermanentStatus

IMPORTANT — The addressing method shall be FUNCTIONAL.

Table 5 — SHORT-NAMES for OBD service 0x01 distinguishing PID 0x13 and PID 0x1D defines two (2) different service 0x01 LONG-NAME and SHORT-NAME because the interpretation of the respective response impacts the interpretation of e.g. PID 0x1B.

Table 5 — SHORT-NAMES for OBD service 0x01 distinguishing PID 0x13 and PID 0x1D

LONG-NAME	SHORT-NAME
Service 0x01 – Request current powertrain diagnostic data (PID 0x13)	Service01RequestCurrentPowertrainDiagnosticDataPID13
Service 0x01 – Request current powertrain diagnostic data (PID 0x1D)	Service01RequestCurrentPowertrainDiagnosticDataPID1D

7.2.3 ODX REQUEST implementation

7.2.3.1 ODX IDs of REQUESTS

For the ID of an OBD REQUEST the prefix Req_ shall be used to distinguish the ID of a REQUEST with the identical ID of a corresponding POS-RESPONSE.

7.2.3.2 LONG-NAMES and SHORT-NAMES of REQUESTS

The LONG-NAME of the REQUEST must be the same as the LONG-NAME of the DIAG-SERVICE that the REQUEST belongs to. The SHORT-NAME of the REQUEST must be the same as the SHORT-NAME of the DIAG-SERVICE that the REQUEST belongs to.

Table 6 — LONG-NAMES and SHORT-NAMES of REQUESTs

LONG-NAME of REQUEST	SHORT-NAME of REQUEST
Service 0x01 – Request current powertrain diagnostic data	Service01RequestCurrentPowertrainDiagnosticData
Service 0x02 – Request powertrain freeze frame data	Service02RequestPowertrainFreezeFrameData
Service 0x03 – Request emission-related diagnostic trouble codes	Service03RequestEmissionRelatedDiagnosticTroubleCodes
Service 0x04 – Clear/reset emission-related diagnostic information	Service04ClearResetEmissionRelatedDiagnosticInformation
Service 0x05 – Request oxygen sensor monitoring test results	Service05RequestOxygenSensorMonitoringTestResults
Service 0x06 – Request on-board monitoring test results for specific monitored systems	Service06RequestOnBoardMonitoringTestResultsForSpecificMonitoredSystems
Service 0x07 – Request emission-related diagnostic trouble codes detected during current or last completed driving cycle	Service07RequestEmissionRelatedDiagnosticTroubleCodesDetectedDuringCurrentOrLastCompletedDrivingCycle
Service 0x08 – Request control of on-board system, test or component	Service08RequestControlOfOnBoardSystemTestOrComponent
Service 0x09 – Request vehicle information	Service09RequestVehicleInformation
Service 0x0A – Request emissions-related diagnostic trouble codes with permanent status	Service0ARequestEmissionRelatedDiagnosticTroubleCodesWithPermanentStatus

7.2.4 ODX POS-RESPONSE implementation

7.2.4.1 ODX IDs of POS-RESPONSES

For the ID of an OBD POS-RESPONSE the prefix Resp_ shall be used to distinguish the ID of a REQUEST with the identical ID of the corresponding POS-RESPONSE.

7.2.4.2 LONG-NAMES of POS-RESPONSES

The LONG-NAME of the POS-RESPONSE must be the same as the LONG-NAME of the DIAG-SERVICE that the POS-RESPONSE belongs to.

7.2.4.3 SHORT-NAMES of POS-RESPONSES

The SHORT-NAME of the POS-RESPONSE must be the same as the SHORT-NAME of the DIAG-SERVICE that the REQUEST belongs to.

Table 7 — LONG-NAMES and SHORT-NAMES of POS-RESPONSES

LONG-NAME of POS-RESPONSE	SHORT-NAME of POS-RESPONSE
Service 0x01 – Request current powertrain diagnostic data	Service01RequestCurrentPowertrainDiagnosticData
Service 0x02 – Request powertrain freeze frame data	Service02RequestPowertrainFreezeFrameData
Service 0x03 – Request emission-related diagnostic trouble codes	Service03RequestEmissionRelatedDiagnosticTroubleCodes
Service 0x04 – Clear/reset emission-related diagnostic information	Service04ClearResetEmissionRelatedDiagnosticInformation
Service 0x05 – Request oxygen sensor monitoring test results	Service05RequestOxygenSensorMonitoringTestResults
Service 0x06 – Request on-board monitoring test results for specific monitored systems	Service06RequestOnBoardMonitoringTestResultsForSpecificMonitoredSystems
Service 0x07 – Request emission-related diagnostic trouble codes detected during current or last completed driving cycle	Service07RequestEmissionRelatedDiagnosticTroubleCodesDetectedDuringCurrentOrLastCompletedDrivingCycle
Service 0x08 – Request control of on-board system, test or component	Service08RequestControlOfOnBoardSystemTestOrComponent
Service 0x09 – Request vehicle information	Service09RequestVehicleInformation
Service 0x0A – Request emissions-related diagnostic trouble codes with permanent status	Service0ArequestEmissionRelatedDiagnosticTroubleCodesWithPermanentStatus

7.3 ODX PARAMs implementation

As a general rule, the ODX names of a parameter shall be derived from the “Description” of the parameter as specified in ISO 15031-5. The LONG-NAME shall be copied from the description in ISO 15031-5. The SHORT-NAME shall be generated by removing all characters from the LONG-NAME disallowed by the ODX specification.

In case a parameter in ISO 15031-5 has no description or when the description is not favourable for a specific parameter, the naming convention will be detailed in this document.

The second parameter at the POS-RESPONSE must describe the PID of the REQUEST. This must be done with the use of MATCHING-REQUEST-PARAM. The LONG-NAME of this parameter is “Matching PID”. The SHORT-NAME must be MatchingParameterID.

7.3.1 Service IDs (SID)

7.3.1.1 LONG-NAMES and SHORT-NAMES of request SIDs

The general rule shall be applied, however the words “request SID” shall be appended to the LONG-NAME of the SID parameter. The SHORT-NAME is then derived from the LONG-NAME as described in the general rule.

Table 8 — LONG-NAMES and SHORT-NAMES of PARAMs of the request SID

Request SID	LONG-NAME of PARAM	SHORT-NAME of PARAM
Service 0x01	Request current powertrain diagnostic data request SID	RequestCurrentPowertrainDiagnosticDataRequestSID
Service 0x02	Request powertrain freeze frame data request SID	RequestPowertrainFreezeFrameDataRequestSID
Service 0x03	Request emission-related diagnostic trouble codes request SID	RequestEmissionRelatedDiagnosticTroubleCodesRequestSID
Service 0x04	Clear/reset emission-related diagnostic information request SID	ClearResetEmissionRelatedDiagnosticInformationRequestSID
Service 0x05	Request oxygen sensor monitoring test results request SID	RequestOxygenSensorMonitoringTestResultsRequestSID
Service 0x06	Request on-board monitoring test results for specific monitored systems request SID	RequestOnBoardMonitoringTestResultsForSpecificMonitoredSystemsRequestSID
Service 0x07	Request emission-related diagnostic trouble codes detected during current or last completed driving cycle request SID	RequestEmissionRelatedDiagnosticTroubleCodesDetectedDuringCurrentOrLastCompletedDrivingCycleRequestSID
Service 0x08	Request control of on-board system, test or component request SID	RequestControlOfOnBoardSystemTestOrComponentRequestSID
Service 0x09	Request vehicle information request SID	RequestVehicleInformationRequestSID

The request SID parameters shall be of type CODED-CONST defined as DIAG-CODED-TYPE, i.e. an 8 bit unsigned integer value.

7.3.1.2 LONG-NAMES and SHORT-NAMES of response SIDs

The general rule shall be applied, however the words “response SID” shall be appended to the LONG-NAME of the SID parameter. The SHORT-NAME is then derived from the LONG-NAME as described in the general rule.

Table 9 — LONG-NAMES and SHORT-NAMES of PARAMs at a service response

LONG-NAME of PARAM	SHORT-NAME of PARAM
Request current powertrain diagnostic data response SID	RequestCurrentPowertrainDiagnosticDataResponseSID
Request powertrain freeze frame data response SID	RequestPowertrainFreezeFrameDataResponseSID
Request emission-related diagnostic trouble codes response SID	RequestEmissionRelatedDiagnosticTroubleCodesResponseSID
Clear/reset emission-related diagnostic information response SID	ClearResetEmissionRelatedDiagnosticInformationResponseSID
Request oxygen sensor monitoring test results response SID	RequestOxygenSensorMonitoringTestResultsResponseSID
Request on-board monitoring test results for specific monitored systems response SID	RequestOnBoardMonitoringTestResultsForSpecificMonitoredSystemsResponseSID
Request emission-related diagnostic trouble codes detected during current or last completed driving cycle response SID	RequestEmissionRelatedDiagnosticTroubleCodesDetectedDuringCurrentOrLastCompletedDrivingCycleResponseSID
Request control of on-board system, test or component response SID	RequestControlOfOnBoardSystemTestOrComponentResponseSID
Request vehicle information response SID	RequestVehicleInformationResponseSID

The response SID parameters shall be of type CODED-CONST defined as DIAG-CODED-TYPE, i.e. an 8 bit unsigned integer value.

7.3.2 Local Identifier implementation

7.3.2.1 General

As a general modelling principle, the services 0x01, 0x02, 0x06, 0x08 and 0x09, shall contain TABLE-KEY parameters. This enforces the definition of request and response message structures in TABLES.

Each TABLE-ROW shall carry a specific semantic as specified in clause 7.3.2.2. This is required to determine which entries in the TABLE are used to inquire diagnostic capabilities and which return actual emissions-related OBD data.

7.3.2.2 SEMANTICS

ODX V2.2.0 supports the specification of TABLE-ROW.SEMANTIC values.

Table 10 — SEMANTIC values for DIAG-SERVICES and TABLEs describes SEMANTIC values for DIAG-SERVICES and TABLEs.

Table 10 — SEMANTIC values for DIAG-SERVICES and TABLEs

Serviceld (PID/OBDMID/TID/ INFOTYPE)	DIAG-SERVICE.SEMANTIC/ TABLE-ROW.SEMANTIC	TABLE.SEMANTIC
Powertrain Diagnostic and Freeze Frame Data		
0x01 (0x00,0x20,...)	READ_DATA_PWRTRAIN_SUPPORTED	PWRTRAIN_CURR
0x01 (0x01,0x02,...)	READ_DATA_PWRTRAIN_CURR	
0x02 (0x00,0x20,...)	READ_DATA_PWRTRAIN_FREEZE_FRAME_SUPPORTED	PWRTRAIN_FREEZE_FRAME
0x02 (0x01,0x02,...)	READ_DATA_PWRTRAIN_FREEZE_FRAME	
Emission-related Trouble Codes		
0x03	READ_DTC_POWERTRAIN	no usage of TABLEs
0x07	READ_DTC_DETECTION_TIME_DEPENDENT	
0x0A	READ_DTC_PERMANENT_STATUS	
0x04	READ_DTC_CLEAR	
Onboard Monitoring Test Results for Specific Monitored Systems		
0x06 (0x00,0x20,...)	READ_OBDMID_SUPPORTED	OBDMID
0x06 (0x01,0x02,...)	READ_OBDMID_DATA	
Control of Onboard Systems, Test or Control		
0x08 (0x00,0x020)	READ_TID_SUPPORTED	TID
0x08 (0x01,0x02)	READ_TID_DATA	
Vehicle Information		
0x09 (0x00,0x20)	READ_INFOTYPE_SUPPORTED	INFOTYPE
0x09 (0x01,0x02,...)	READ_INFOTYPE_DATA	

7.3.3 Service-specific parameters

7.3.3.1 Service-specific request parameters for services 0x01, 0x02, 0x06, 0x08 and 0x09

Parameters 2 and above are collected in an END-OF-PDU-FIELD Parameter. For services 0x01, 0x06, 0x08 and 0x09 an END-OF-PDU-FIELD with MAX-NUMBER-OF-ITEMS equal to 6 and MIN-NUMBER-OF-ITEMS equal to 1 shall be used. The SHORT-NAME of this END-OF-PDU-FIELD shall be "PIDsSupportedServices01060809RequestPID13" and "PIDsSupportedServices01060809RequestPID1D" respectively. For service 0x02 an END-OF-PDU-FIELD with MAX-NUMBER-OF-ITEMS equal to 3 and MIN-NUMBER-OF-ITEMS equal to 1 shall be used. The SHORT-NAME of this END-OF-PDU-FIELD shall be "PIDsSupportedServices02Request"

The END-OF-PDU-FIELD "PIDsSupportedServices01060809RequestPID13" references an ODX STRUCTURE with SHORT-NAME "PIDsSupportedRequestPID13".

"PIDsSupportedRequestPID13" has a single PARAM of Type TABLE-KEY with SHORT-NAME "PIDRequest". "PIDRequest" references an object of type TABLE with SHORT-NAME "PIDsSupportedForPID13". The TABLE has one TABLE-ROW for each PID (00 – FF) with LONG-NAME "OBD Supported ID <PIDnumber>" and SHORT-NAME "OBDSupportedID<PIDnumber>". The corresponding TABLE-STRUCT shall also be used when decoding the response message (see below).

The value of the TABLE-KEY shall be the value of the PID.

END-OF-PDU-FIELD “PIDsSupportedServiceS02” reference an ODX STRUCTURE with SHORT-NAME “PIDsFramesSupported”. “PIDsFramesSupported” shall have two parameters: The first parameter is of type TABLE-KEY with SHORT-NAME “PIDs”. “PIDs” references an object of type TABLE with SHORT-NAME “PIDsSupported”. The second parameter is of type VALUE and its SHORT-NAME shall be “FrameNo”. The application using this information may then store the requested frame number in this parameter before sending the request.

Figure 8 — Service 0x01 - Request current powertrain diagnostic data (PID 0x13=1) shows a graphical representation of the ODX service definition structure taking service 0x01 as an example. The diagram also includes the data elements used in the response definition.

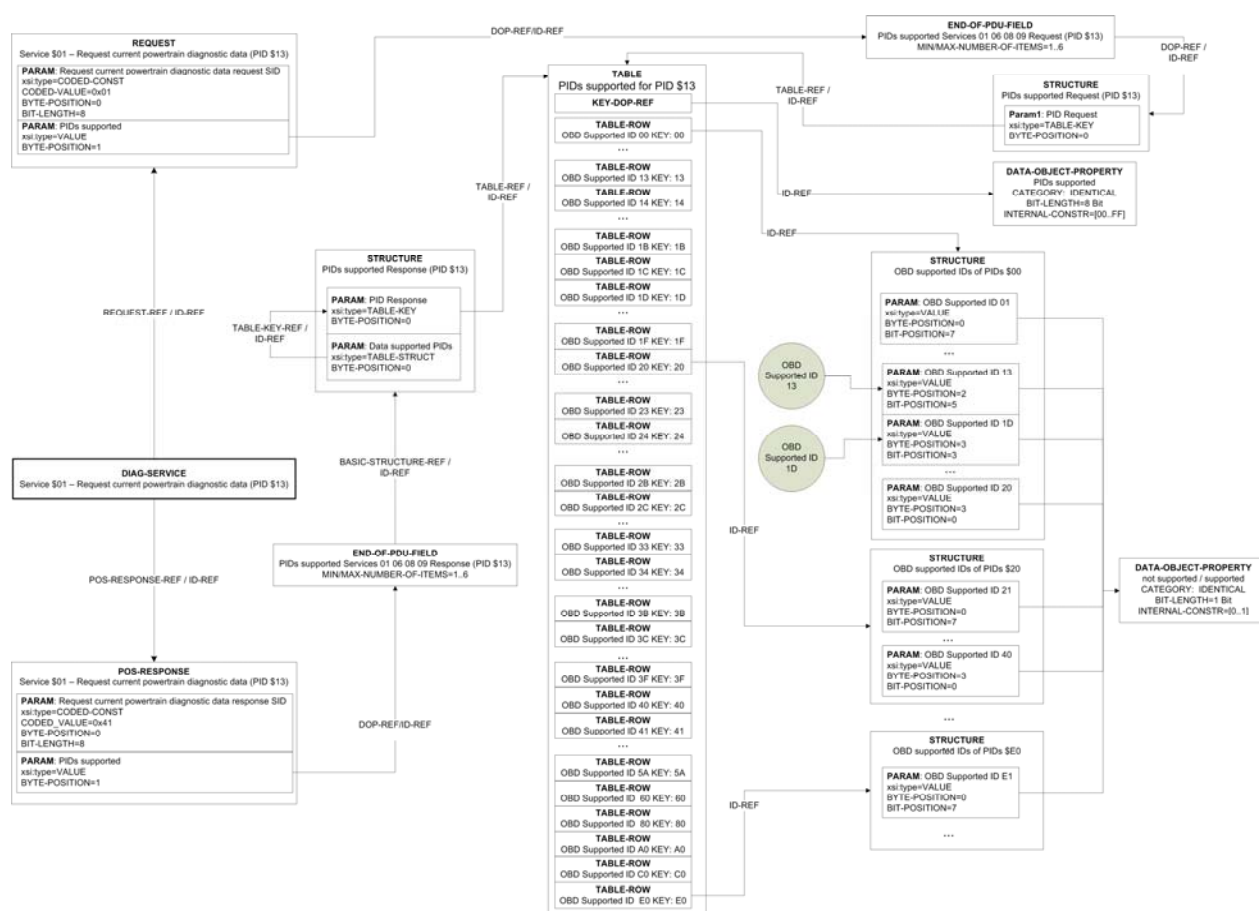


Figure 8 — Service 0x01 - Request current powertrain diagnostic data (PID 0x13=1)

In an ECU implementation only either PID 0x13 or PID 0x1D are supported as mandated by ISO 15031-5. Different response messages are defined for each case.

In case PID 0x13 is supported, the following regulation applies:

The SUB-COMPONENT “PIDs 0x14 to 0x1B and 0x24 to 0x2B and 0x34 to 0x3B for PID 0x13 equal to 1” defines which service has to be used and which TABLE has to be referenced to get the correct data definitions within the response. This case is shown with the next figure.

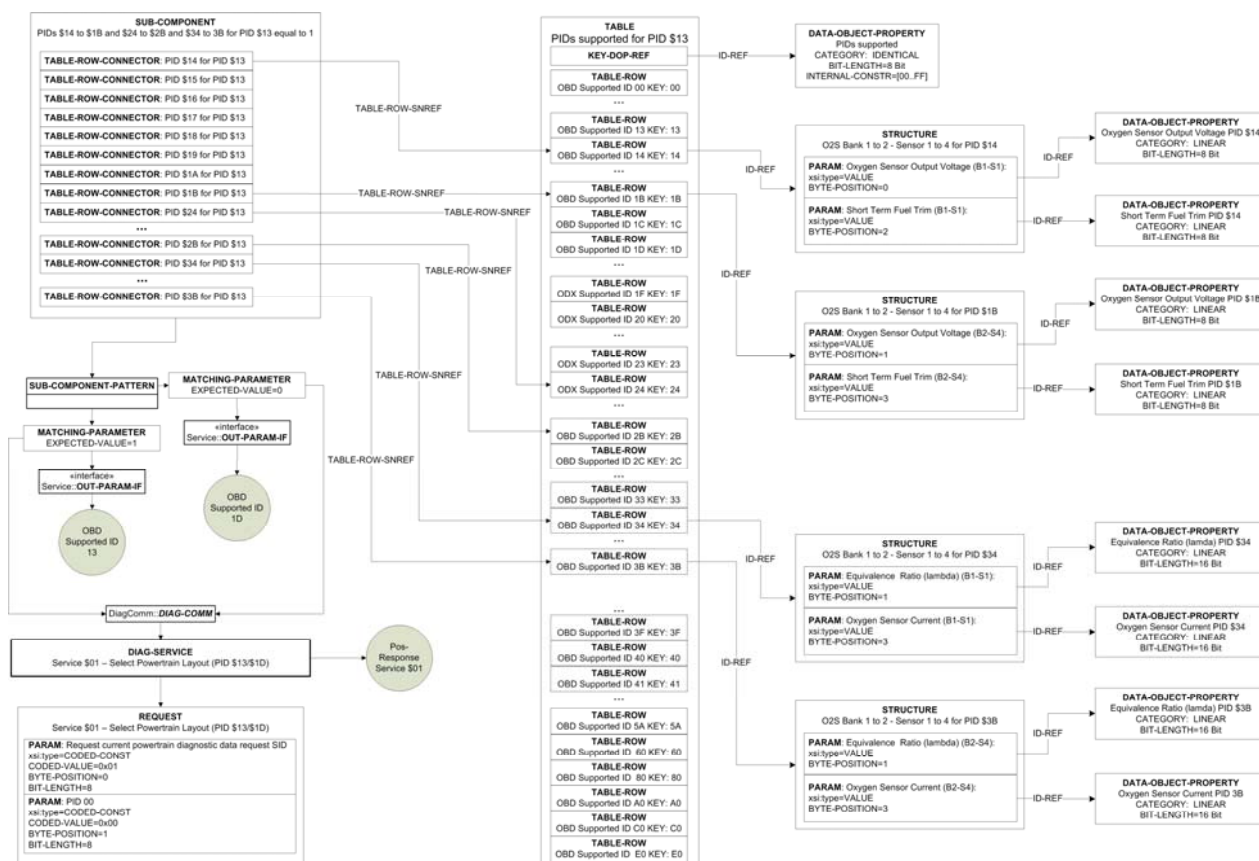


Figure 9 — SUB-COMPONENT for the use with service 0x01 and PID 0x13=1 and PID 0x1D=0

In case PID 0x1D is supported, the following rule applies:

The SUB-COMPONENT “PIDs 0x14 to 0x1B and 0x24 to 0x2B and 0x34 to 0x3B for PID 0x1D equal to 1” defines which service has to be used and which TABLE has to be referenced to get the correct data definitions within the response. This case is shown with the next figure.

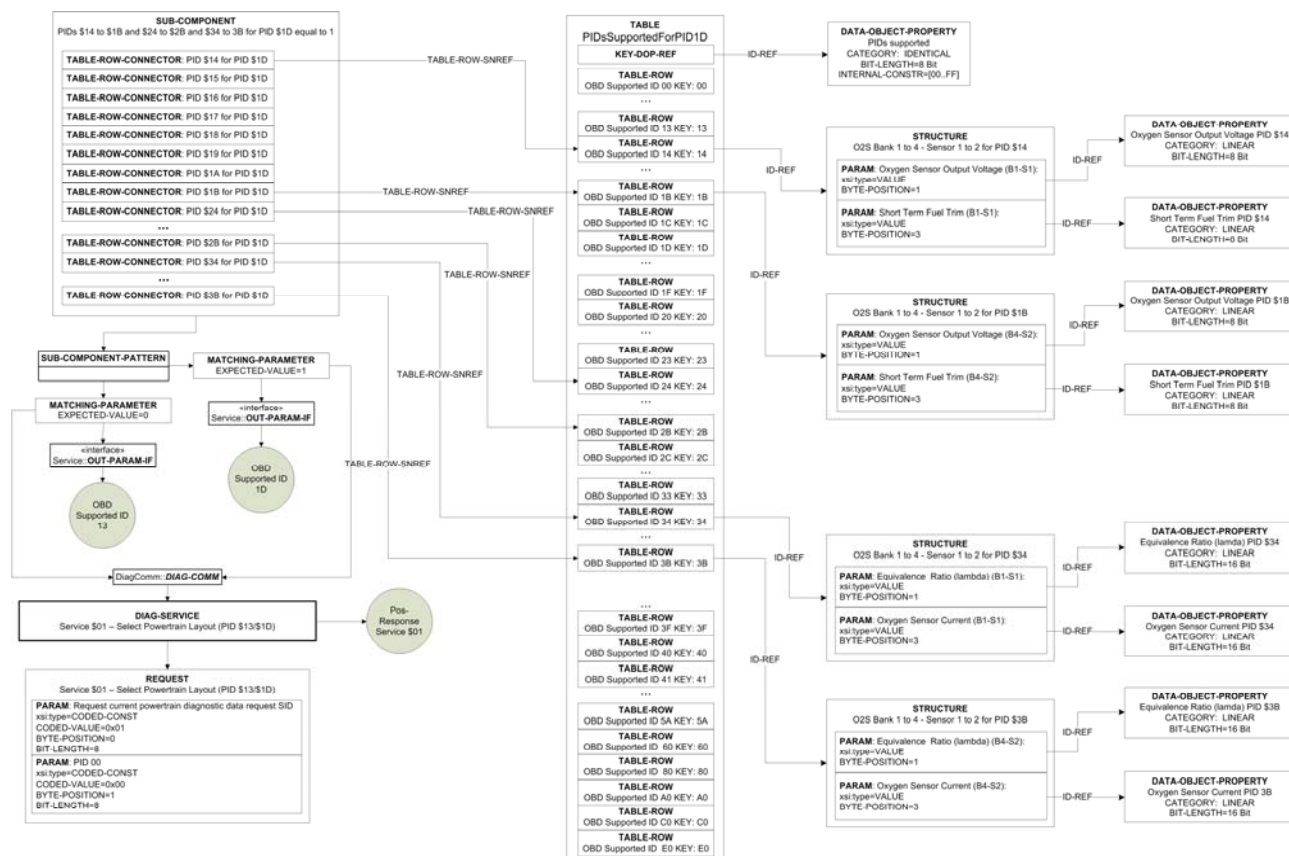


Figure 10 — SUB-COMPONENT for the use with service 0x01 and PID 0x13=0 and PID 0x1D=1

7.3.3.2 Service-specific request parameters for services 0x03 0x04 and 0x07

For services 0x03, 0x04 and 0x07 no service-specific request parameters are defined because these services take no further parameters beside the SID.

7.3.3.3 Service-specific response parameters for services 0x01, 0x08 and 0x09

The response message shall use the same data structures as the corresponding request message, i.e. the TABLE with SHORT-NAME “PIDsSupported” (see above). The second response parameter returns the requested PID. In the ODX representation, parameters 2 and above are collected in an END-OF-PDU-FIELD “PIDsSupported01060809Response” parameter similar to the request message. The ODX STRUCTURE corresponding to “PIDsSupported01060809Response” shall reference the TABLE-KEY of TABLE “PIDsSupported” in its first parameter and TABLE-STRUCT of TABLE “PIDsSupported” in its second parameter.

7.3.3.4 Service-Specific Response Parameters for Services 0x02

The response message shall have 2 parameters. The naming of parameter one follows the rule for Service ID explained earlier. The second response parameter shall be an END-OF-PDU-FIELD.

7.3.3.5 Service-Specific Response Parameters for Services 0x03 and 0x07

The response message shall have 3 parameters. The naming of parameter one follows the rule for Service ID explained earlier. The second response parameter, called “NoDTCS” of type VALUE, returns the number of stored DTCs. The third parameter shall be of type END-OF-PDU-FIELD called “StoredDTCS”.

7.3.3.6 Service-Specific Response Parameters for services 0x04

The response message shall have one parameter. The naming of parameter one follows the rule for Service ID explained earlier. For a positive response, there are no further parameters defined in ISO 15031-5.

7.3.3.7 Service-Specific Response Parameters for services 0x06

The response message shall have 3 parameters. The naming of parameter one follows the rule for Service ID explained earlier. The second response parameter shall be of type END-OF-PDU-FIELD.

7.4 Conversion of PIDs to ODX

7.4.1 Properties of PIDs

Rules for authoring of PIDs:

- Rule 0: PIDs are described in ISO[®]15031-5 in tables with columns PID, Description, Data byte, Scaling/bit and “External test equipment SI (Metric) / English display” and optionally min and max value. A value is a line in a table where a bit or a bit range is specified in column “Data byte”.
- Rule 1: The value(s) of the same PID are mapped to one ODX STRUCTURE.

For the purpose of mapping the values of a PID to ODX, these values are categorized in a number of different types. See Table 11 — Value Types.

Table 11 — Value Types

Value Type	Description	Example Value
Linear	The value describes a continuous numeric range. In ISO [®] 15031-5 typically min value, max value, scaling and a unit are given.	PID 0x06 Byte A
Boolean	The value refers to a single bit, such as an attribute like ON/OFF or YES/NO	PID 0x01 Byte A Bit 7
BitSelect	The values cover a range of bits of which at most one is set. The result depends on which bit is set.	PID 0x03 Byte A
BitSet	The values cover a range of bits. Each bit represents a property to be present or absent.	PID 0x13
Number2Text	The value covers a range of bits. The bits are interpreted as a number and each number translates to some text.	PID 0x1C
Number	The value covers a range of bits. The bits are directly interpreted as a number without conversion.	PID 0x01 Byte A Bits 0 - 6
DTC	The value covers 2 bytes and is interpreted as a diagnostic trouble code.	PID 0x02

7.4.2 Rules for type “Linear”

"Linear" rules:

- Rule L1: Each value is mapped to a parameter of type VALUE.
- Rule L2: The LONG-NAME of the parameter shall be the content of the column “Description” of the corresponding value.
- Rule L3: The attribute TI of the LONG-NAME shall be “OBD_PID<#>_<SHORT-NAME>”.
- Rule L4: The SHORT-NAME shall be generated from the LONG-NAME by the following algorithm: All characters disallowed in an ODX SHORT-NAME shall be removed. Wherever there is a blank in the LONG-NAME, the next character in the SHORT-NAME shall be converted to upper case.
- Rule L5: The BYTE-POSITION depends on the “Data byte” in the table. For byte A, B, ... the BYTE-POSITION shall be 0, 1, ... respectively.
- Rule L6: The parameter shall reference a DOP with DIAG-CODED-TYPE of type STANDARD-LENGTH-TYPE. Its BIT-LENGTH shall be set according to the “Data byte” column, typically 8 or 16. The COMPU-METHOD shall be of CATEGORY “LINEAR”. The COMPU-RATIONAL-COEFFS shall be set according to the “Scaling/bit” column. A UNIT shall be referenced that describes the unit in the max value column. The physical type shall be represented as A_UINT32 if only positive integer values can occur, as A_INT32 if integers can occur and as A_FLOAT otherwise. The PRECISION shall be set according to the metric value in the column “External test equipment SI (Metric) / English display”.

7.4.3 Rules for type “Boolean”

"Boolean" rules:

- Rule O1: Each value is mapped to a parameter of type VALUE.
- Rule O2: The LONG-NAME of the parameter shall be the content of the column “Description” of the corresponding value.
- Rule O3: The attribute TI of the LONG-NAME shall be “OBD_PID<#>_<SHORT-NAME>”.
- Rule O4: The SHORT-NAME shall be generated from the LONG-NAME by the following algorithm: All characters disallowed in an ODX SHORT-NAME shall be removed. Wherever there is a blank in the LONG-NAME, the next character in the SHORT-NAME shall be converted to upper case.
- Rule O5: The BYTE-POSITION depends on the “Data byte” in the table. For byte A, B, ... the BYTE-POSITION shall be 0, 1, ... respectively.
- Rule O6: The BIT-POSITION shall be determined by the value in “Data byte”.
- Rule O7: The parameter shall reference a DOP with DIAG-CODED-TYPE of type STANDARD-LENGTH-TYPE. Its BIT-LENGTH shall be set to 1. The COMPU-METHOD shall be of CATEGORY “TEXTTABLE”. The VT values of this TEXTTABLE shall be set according to the entries in column “External test equipment SI (Metric) / English display”. LOWER-LIMIT shall be set according to the entries in column “Scaling/bit”.

7.4.4 Rules for type “BitSelect”

"BitSelect" rules:

- Rule B1: All values are mapped to a single parameter of type VALUE.
- Rule B2: The LONG-NAME of the parameter shall be the content of the column “Description” of the corresponding Byte (e.g. “Fuel system 1 status:” for PID 0x03 Byte A).
- Rule B3: The attribute TI of the LONG-NAME shall be “OBD_PID<#>_<SHORT-NAME>”.
- Rule B4: The SHORT-NAME shall be generated from the LONG-NAME by the following algorithm: All characters disallowed in an ODX SHORT-NAME shall be removed. Wherever there is a blank in the LONG-NAME, the next character in the SHORT-NAME shall be converted to upper case.
- Rule B5: The BYTE-POSITION depends on the “Data byte” in the table. For byte A, B, ... the BYTE-POSITION shall be 0, 1, ... respectively.
- Rule B6: The parameter shall reference a DOP with DIAG-CODED-TYPE of type STANDARD-LENGTH-TYPE. Its BIT-LENGTH shall be set to 8. The COMPU-METHOD shall be of CATEGORY “TEXTTABLE”. The TEXTTABLE shall contain one entry for each non-reserved value. LOWER-LIMIT shall be set to 2 to the power of bit position, i. e. 1, 2, 4, ... The VT values of this TEXTTABLE shall be set according to the entries in column “External test equipment SI (Metric) / English display”. An additional entry shall be created with LOWER-LIMIT 0 and VT equal to “-” if it is valid that no bit is set.

7.4.5 Rules for type “BitSet”

"BitSet" rules:

- Rule S1: Each value is mapped to a parameter of type VALUE.
- Rule S2: The LONG-NAME of each parameter shall be the concatenation of the content of the column “Description” of the corresponding Byte (e. g. “Location of Oxygen Sensors” for PID 0x13 Bit 0 and thus Byte A) and the deterministic part of the content of the column “Scaling/bit” of the corresponding value (e. g. “Bank 1 Sensor 1” for PID 0x13 Bit 0) separated by space.
- Rule S3: The attribute TI of the LONG-NAME shall be “OBD_PID<#>_<SHORT-NAME>”.
- Rule S4: The SHORT-NAME shall be generated from the LONG-NAME by the following algorithm: All characters disallowed in an ODX SHORT-NAME shall be removed. Wherever there is a blank in the LONG-NAME, the next character in the SHORT-NAME shall be converted to upper case.
- Rule S5: The BYTE-POSITION depends on the “Data byte” in the table. For byte A, B, ... the BYTE-POSITION shall be 0, 1, ... respectively.
- Rule S6: The BIT-POSITION shall be determined by the value in “Data byte”.
- Rule S7: The parameter shall reference a DOP with DIAG-CODED-TYPE of type STANDARD-LENGTH-TYPE. Its BIT-LENGTH shall be set to 1. The COMPU-METHOD shall be of CATEGORY “TEXTTABLE”. 1 shall be mapped to the text shown in column “External test equipment SI (Metric) / English display”. 0 shall be mapped to the empty text.

7.4.6 Rules for type “Number2Text”

"Number2Text" rules:

- Rule X1: The value is mapped to a parameter of type VALUE.
- Rule X2: The LONG-NAME of the parameter shall be the content of the column “Description” of the corresponding byte (e. g. “OBD requirements to which vehicle is designed”).
- Rule X3: The attribute TI of the LONG-NAME shall be “OBD_PID<#>_<SHORT-NAME>”.
- Rule X4: The SHORT-NAME shall be generated from the LONG-NAME by the following algorithm: All characters disallowed in an ODX SHORT-NAME shall be removed. Wherever there is a blank in the LONG-NAME, the next character in the SHORT-NAME shall be converted to upper case.
- Rule X5: The BYTE-POSITION depends on the referenced data byte. For byte A, B, ... the BYTE-POSITION shall be 0, 1, ... respectively.
- Rule X6: The parameter shall reference a DOP with DIAG-CODED-TYPE of type STANDARD-LENGTH-TYPE. Its BIT-LENGTH shall be set to the number of bits covered by the value, typically 8. The COMPU-METHOD shall be of CATEGORY “TEXTTABLE. The entries in column “Data byte” shall be mapped to the texts in column “External test equipment SI (Metric) / English display”.

NOTE ODX uses the decimal representation of numbers.

7.4.7 Rules for type “Number”

"Number" rules:

- Rule N1: The value is mapped to a parameter of type VALUE.
- Rule N2: The LONG-NAME of the parameter shall be the content of the column “Description” of the corresponding value.
- Rule N3: The attribute TI of the LONG-NAME shall be “OBD_PID<#>_<SHORT-NAME>”.
- Rule N4: The SHORT-NAME shall be generated from the LONG-NAME by the following algorithm: All characters disallowed in an ODX SHORT-NAME shall be removed. Wherever there is a blank in the LONG-NAME, the next character in the SHORT-NAME shall be converted to upper case.
- Rule N5: The BYTE-POSITION depends on the “Data byte” in the table. For byte A, B, ... the BYTE-POSITION shall be 0, 1, ... respectively.
- Rule N6: If the column “Data byte” describes a bit range, the BIT-POSITION shall be set to the lower value. If zero the BIT-POSITION may be omitted.
- Rule N7: The parameter shall reference a DOP with DIAG-CODED-TYPE of type STANDARD-LENGTH-TYPE. Its BIT-LENGTH shall be set according to the “Data byte” column. The COMPU-METHOD shall be of CATEGORY “IDENTICAL”. The DISPLAY-RADIX shall be set according to the entry in column “Scaling/bit”.

7.4.8 Rules for type “DTC”

"DTC" rules:

- Rule D1: The value is mapped to a parameter of type VALUE.
- Rule D2: The LONG-NAME of the parameter shall be the content of the column “Description” of the corresponding byte (e. g. “OBD requirements to which vehicle is designed”).
- Rule D3: The attribute TI of the LONG-NAME shall be “OBD_PID<#>_<SHORT-NAME>”.
- Rule D4: The SHORT-NAME shall be generated from the LONG-NAME by the following algorithm: All characters disallowed in an ODX SHORT-NAME shall be removed. Wherever there is a blank in the LONG-NAME, the next character in the SHORT-NAME shall be converted to upper case.
- Rule D5: The BYTE-POSITION depends on the referenced data byte. For byte A, B, ... the BYTE-POSITION shall be 0, 1, ... respectively.
- Rule D6: The parameter shall reference a DTC-DOP with SHORT-NAME “ObdDtcs”.

7.4.9 Special handling of PID 0x13 and 0x1D

Depending on the settings of PIDs 0x13 and 0x1D a different response behaviour is required. In case the service request “01 00” shows “supported” for the PID 0x13 in the response and “not supported” for the PID 0x1D the PIDs 0x14 to 0x1B need to be interpreted as shown in Table B.21 – PID 0x14 - 0x1B definition in ISO 15031-5. In the opposite case the PIDs 0x14 to 0x1B need to be interpreted as shown in Table B.22 – PID 0x14 - 0x1B definition in ISO 15031-5.

To express this behaviour in ODX the element SUB-COMPONENT in DIAG-LAYER “OBDII Common Services” is used.

The SUB-COMPONENT with SHORT-NAME “PIDs14To1BAnd24To2BAnd34To3BForPID13EqualTo1” for PID 0x13 “supported” is shown in the following figure.

[illegible]

Figure 11 — SUB-COMPONENT for PID 0x13 supported

If PID 0x13 is supported, a tester application shall use the information of the SUB-COMPONENT with the LONG-NAME "PIDs 0x14 to 0x1B and 0x24 to 0x2B and 0x34 to 3B for PID 0x13 equal to 1" for evaluation of the response when requesting the PIDs 0x14 to 0x1B, 0x24 to 0x2B, 0x34 to 0x3B. The TABLE-REF/ID-REF of this SUB-COMPONENT shows where to look for the response definition of the PIDs.

If PID 0x1D is supported, a tester application shall use the information of the SUB-COMPONENT with the LONG-NAME “PIDs 0x14 to 0x1B and 0x24 to 0x2B and 0x34 to 3B for PID 0x1D equal to 1” for evaluation of the response when requesting the PIDs 0x14 to 0x1B, 0x24 to 0x2B, 0x34 to 0x3B. The TABLE-REF/ID-REF of this SUB-COMPONENT shows where to look for the response definition of the PIDs.

SUB-COMPONENT	
ID	SubComp_PIDs14To1BAnd24To2BAnd34To3BForPID1DEqualTo1
SHORT-NAME	PIDs14To1BAnd24To2BAnd34To3BForPID1DEqualTo1
LONG-NAME	PIDs \$14 to \$1B and \$24 to \$2B and \$34 to 3B for PID \$1D equal to 1
SUB-COMPONENT-PATTERNS	
SUB-COMPONENT-PATTERN	
MATCHING-PARAMETERS	
MATCHING-PARAMETER	
EXPECTED-VALUE	0
DIAG-COMM-SHREF	SHORT-NAME Service01RequestCurrentPowertrainDiagnosticDataPID1D
OUT-PARAM-IF-SHREF	SHORT-NAME OBDSupportedID13
MATCHING-PARAMETER	
EXPECTED-VALUE	1
DIAG-COMM-SHREF	SHORT-NAME Service01RequestCurrentPowertrainDiagnosticDataPID1D
OUT-PARAM-IF-SHREF	SHORT-NAME OBDSupportedID1D
TABLE-ROW-CONNECTORS	
TABLE-ROW-CONNECTOR	
SHORT-NAME	PID14ForPID1D
LONG-NAME	PID \$14 for PID \$1D
TABLE-REF	ID-REF ES_OBDIICommonServices.TAB_PIDsSupportedForPID1D
TABLE-ROW-SHREF	SHORT-NAME OBDSupportedID14
TABLE-ROW-CONNECTOR	
TABLE-ROW-CONNECTOR	
TABLE-ROW-CONNECTOR	
TABLE-ROW-CONNECTOR	
TABLE-ROW-CONNECTOR	
TABLE-ROW-CONNECTOR	
TABLE-ROW-CONNECTOR	

Figure 12 — SUB-COMPONENT for PID 0x1D supported

The same behaviour as above is given with the service request “01 20”. With the dependence of the settings of PIDs 0x13 and 0x1D the PIDs 0x24 to 0x2B need to be treated as shown in Table B.30 – PID 0x24 - 0x2B definition respectively as shown in Table B.31 – PID 0x24 - 0x2B definition in ISO°15031-5.

The same behaviour as above is given with the service request “01 20”. With the dependence of the settings of PIDs 0x13 and 0x1D the PIDs 0x34 to 0x3B need to be treated as shown in Table B.40 – PID 0x34 - 0x3B definition respectively as shown in Table B.41 – PID 0x34 - 0x3B definition in ISO°15031-5.

7.5 Conversion of DTCs to ODX

While for services ISO°15031 already defines the complete set, for e. g. DTCs it is expected that the currently defined set will be extended in the near future. For that reason, the concrete data for diagnostic trouble codes is defined in the digital annexes of ISO°15031. This standard describes rules to derive concrete ODX data from these annexes.

All DTC elements are to be defined in the ECU-SHARED-DATA diagnostic layer with short name ObdIIDopsDtcDeclarations and ID ES_ObdIIDopsDtcDeclarations. It may contain any LONG-NAME. It shall contain a DIAG-DICTIONARY-SPEC. In addition it may only contain ADMIN-DATA, COMPANY-DATAS, and SDGS. It shall not contain any other element.

The DIAG-DICTIONARY-SPEC shall contain DTC-DOPS and may contain ADMIN-DATA, and SDGS. It shall not contain any other element.

ISO°15031-6 defines four groups of DTCs: Powertrain (their names start with P), Chassis (their names start with C), Body (their names start with B), and Network (their name starts with U). The DIAG-DICTIONARY-SPEC contains a single DTC-DOP for each of these four groups. Each DTC-DOP lists all the DTCs defined for that group in ISO°15031-6.

Their IDs and SHORT-NAMEs shall be defined as in Table 12 — SHORT-NAME and ID of DTC-DOPs.

Table 12 — SHORT-NAME and ID of DTC-DOPs

Marker	Group	SHORT-NAME of DTC-DOP	ID attribute of DTC-DOP
P	Powertrain	ObdDtcPowertrain	ES_ObdIIDopsDtcDeclarations.DOP_ObdDtcPowertrain
C	Chassis	ObdDtcChassis	ES_ObdIIDopsDtcDeclarations.DOP_ObdDtcChassis
B	Body	ObdDtcBody	ES_ObdIIDopsDtcDeclarations.DOP_ObdDtcBody
U	Network	ObdDtcNetwork	ES_ObdIIDopsDtcDeclarations.DOP_ObdDtcNetwork

Each DTC-DOP shall contain the same DIAG-CODED-TYPE, PHYSICAL-TYPE, and COMPU-METHOD as shown below:

```
<DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
  <BIT-LENGTH>16</BIT-LENGTH>
</DIAG-CODED-TYPE>
<PHYSICAL-TYPE BASE-DATA-TYPE="A_UINT32" DISPLAY-RADIX="HEX" />
<COMPU-METHOD>
  <CATEGORY>IDENTICAL</CATEGORY>
</COMPU-METHOD>
```

In addition it shall contain a DTCS element that lists all the diagnostic trouble codes defined in the digital annex of ISO°15031 for the corresponding group (Powertrain, Chassis, Body, and Network). The digital annexes list the following fields for each DTC: Number (e. g. P000A), DTC Description (e. g. "A" Camshaft Position Slow Response), Location (e. g. Bank 1), and comment. Footnote marks like "a" are not be considered as content of the corresponding fields. These fields shall be transformed into ODX elements as follows:

- A DTC element shall exist for each DTC line in the annex. Its ID attribute shall have the prefix `ObdDtc_` immediately followed by the value of the Number field of the digital annex, for example: `<DTC ID="ObdDtc_P000A">`. It must not contain the IS-TEMPORARY attribute but may contain the OID attribute with an arbitrary value. The element shall contain the elements SHORT-NAME, TROUBLE-CODE, DISPLAY-TROUBLE-CODE, and TEXT in that order.
- The SHORT-NAME element shall contain the same value as the ID attribute of the DTC element itself, for example: `<SHORT-NAME>ObdDtc_P000A</SHORT-NAME>`
- The TROUBLE-CODE element shall contain a decimal representation of the Number field's content. This number is defined by reverting the algorithm described in SAE J2012. The last four digits are interpreted as a hexadecimal number. Depending on the group the hexadecimal value of Table 13 — Value offset for different DTC groups is added:

Table 13 — Value offset for different DTC groups

Group	Value to add
Powertrain	0x0000
Chassis	0x4000
Body	0x8000
Network	0xC000

The result as a decimal value is the content of the TROUBLE-CODE element. For example P000A is converted as $0x000A + 0x0000 = 0xA = 10$ and results in `<TROUBLE-CODE>10</TROUBLE-CODE>`, while B1001 is converted as $0x1001 + 0x8000 = 0x9001 = 36865$ and, thus, `<TROUBLE-CODE>36865</TROUBLE-CODE>`.

- The DISPLAY-TROUBLE-CODE element shall contain the Number field's value, e. g. `<DISPLAY-TROUBLE-CODE>P000A</DISPLAY-TROUBLE-CODE>`.
- The TEXT element shall contain the concatenation of the Description field's and the Location field's value of the DTC separated by a space, e. g. `<TEXT>"A" Camshaft Position Slow Response Bank 1</TEXT>`. The value or existence of a TI attribute is not defined by this standard.
- If the value of the comment field is added it shall be inserted as an XML comment after the TEXT element. If the comment contains "--" it shall be replaced by a single dash "-" as XML comments must not contain "--".

7.6 ODX samples of ISO^o15031-5 services and authored data

7.6.1 General

In ODX, services with their request and their response(s) are defined with the ODX elements DIAG-SERVICE, REQUEST and POS-RESPONSE (more than one are allowed). Within the definition of the DIAG-SERVICE the references to the REQUEST and the POS-RESPONSE are itemized.

The DIAG-SERVICE defines the LONG-NAME and the SHORT-NAME of the service itself as it is described in clause 7.2.2.3 and 7.2.2.4. The references to the appropriate request and to the appropriate response(s) is done with the ID-REF within the elements REQUEST-REF and POS-RESPONSE-REF. The ID-REFs are filled with the ID of the REQUEST and the POS-RESPONSE respectively.

The following ISO^o15031-5 examples will show the structure of those request and response messages.

7.6.2 Service 0x01 — Request current powertrain diagnostic data

7.6.2.1 ISO 22900-2 — DIAG-SERVICE 0x01 with LONG-NAME and SHORT-NAME sample

```
<DIAG-SERVICE ID="DiagnServi_Service01RequestCurrentPowertrainDiagnosticData
  ADDRESSING="FUNCTIONAL">
  <SHORT-NAME>Service01RequestCurrentPowertrainDiagnosticData</SHORT-NAME>
  <LONG-NAME>Service 0x01 - Request current powertrain diagnostic data</LONG-NAME>
  <REQUEST-REF ID-REF="Req_Service01RequestCurrentPowertrainDiagnosticData"/>
  <POS-RESPONSE-REFS>
    <POS-RESPONSE-REF
      ID-REF="Resp_Service01RequestCurrentPowertrainDiagnosticData"/>
    </POS-RESPONSE-REFS>
</DIAG-SERVICE>
```

7.6.2.2 ISO¹⁵⁰³¹⁻⁵ — Read supported PIDs

The purpose of this service is to allow access to current emission-related data values, including analogue inputs and outputs, digital inputs and outputs, and system status information. The request for information includes a parameter identification (PID) value that indicates to the on-board system the specific information requested.

Not all PIDs are applicable or supported by all systems. PID 0x00 is a bit-encoded value that indicates for each ECU which PIDs are supported. PID 0x00 indicates support for PIDs from 0x01 to 0x20. PID 0x20 indicates support for PIDs 0x21 through 0x40, etc. This is the same concept for PIDs/OBD Monitor IDs/TIDs/InfoTypes support in Services 0x01, 0x02, 0x06, 0x08, 0x09.

EXAMPLE The external test equipment requests supported PIDs (0x00, 0x20, 0x40, 0x60, 0x80, 0xA0) from the vehicle. ECU(s) shall respond to all supported ranges if requested. A range is defined as a block of 32 PIDs (e.g. range #1: PID 0x01-0x20). The ECU shall not respond to unsupported PID ranges unless subsequent ranges have a supported PID(s).

Table 14 — Request current powertrain diagnostic data request message

Message direction:	External test equipment → All ECUs		
Message Type:	Request		
Data Byte	Description (All PID values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Request current powertrain diagnostic data request SID	0x01	SIDRQ
#2	PID used to determine PID support for PIDs 0x01-0x20	0x00	PID
#3	PID used to determine PID support for PIDs 0x21-0x40	0x20	PID
#4	PID used to determine PID support for PIDs 0x41-0x60	0x40	PID
#5	PID used to determine PID support for PIDs 0x61-0x80	0x60	PID
#6	PID used to determine PID support for PIDs 0x81-0xA0	0x80	PID
#7	PID used to determine PID support for PIDs 0xA1-0xC0	0xA0	PID

Table 15 — ECU#1 response: Request current powertrain diagnostic data response message

Message direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	Description (All PID values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Request current powertrain diagnostic data response SID	0x41	SIDPR
#2	PID requested	0x00	PID
#3	Data byte A, representing support for PIDs 0x01, 0x03-0x08	10111111b = 0xBF	DATA_A
#4	Data byte B, representing support for PIDs 0x09, 0x0B-0x10	10111111b = 0xBF	DATA_B
#5	Data byte C, representing support for PIDs 0x11, 0x13, 0x15	10101000b = 0xA8	DATA_C
#6	Data byte D, representing support for PIDs 0x19, 0x1C, 0x20	10010001b = 0x91	DATA_D
#7	PID requested	0x20	PID
#8	Data byte A, representing support for PID 0x21	10000000b = 0x80	DATA_A
#9	Data byte B, representing no support for PIDs 0x29-0x30	00000000b = 0x00	DATA_B
#10	Data byte C, representing no support for PIDs 0x31-0x38	00000000b = 0x00	DATA_C
#11	Data byte D, representing no support for PIDs 0x39-0x40	00000000b = 0x00	DATA_D

7.6.2.3 ISO 22900-2 — Read supported PIDs ODX sample

The request and response message described in this section can be used within ISO 15765-4.

An ODX example of above request and response message is provided below.

```
<REQUEST ID="Req_Service01RequestCurrentPowertrainDiagnosticData">
  <SHORT-NAME>Service01RequestCurrentPowertrainDiagnosticData</SHORT-NAME>
  <LONG-NAME>Service 0x01 - Request current powertrain diagnostic data</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="CODED-CONST">
      <SHORT-NAME>RequestCurrentPowertrainDiagnosticDataRequestSID</SHORT-NAME>
      <LONG-NAME>Request current powertrain diagnostic data request SID</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <CODED-VALUE>1</CODED-VALUE>
      <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
        <BIT-LENGTH>8</BIT-LENGTH>
      </DIAG-CODED-TYPE>
    </PARAM>
    <PARAM xsi:type="VALUE">
      <SHORT-NAME>PIDsSupported</SHORT-NAME>
      <LONG-NAME>PIDs supported</LONG-NAME>
      <BYTE-POSITION>1</BYTE-POSITION>
      <DOP-REF ID-REF="EOPDUF_PIDsSupportedServices01060809Request"/>
    </PARAM>
  </PARAMS>
</REQUEST>
```

The PARAM “PIDsSupported” of the REQUEST “Service01RequestCurrentPowertrainDiagnosticData” with the ID=“Req_Service01RequestCurrentPowertrainDiagnosticData” allows the user to define up to six requesting PIDs (00, 20, 40, .. C0) to find out the appropriate supported PIDs belonging to the requesting PIDs.

The requesting PIDs are defined in the TABLE “PIDsSupported”. The reference to this TABLE is established via the PARAM “PIDsSupported”.

```
<TABLE ID="ES_OBDIICommonServices.TAB_PIDsSupported">
  <SHORT-NAME>PIDsSupported</SHORT-NAME>
  <LONG-NAME>PIDs supported</LONG-NAME>
  <KEY-DOP-REF ID-REF="ES_OBDIICommonServices.DOP_PIDsSupported"/>
  <TABLE-ROW ID="ES_OBDIICommonServices.TAB_PIDsSupported.TABROW_OBDSupportedID00">
    <SHORT-NAME>OBDSupportedID00</SHORT-NAME>
    <LONG-NAME>OBD Supported ID 00</LONG-NAME>
    <KEY>00</KEY>
    <STRUCTURE-REF ID-REF="STRUC_OBDSupportedIDsOfPIDs00And20ToE0"/>
  </TABLE-ROW>

  ...

  <TABLE-ROW ID="ES_OBDIICommonServices.TAB_PIDsSupported.TABROW_OBDSupportedID20">
    <SHORT-NAME>ODXSupportedID20</SHORT-NAME>
    <LONG-NAME>ODX Supported ID 20</LONG-NAME>
    <KEY>20</KEY>
    <STRUCTURE-REF ID-REF="STRUC_OBDSupportedIDsOfPIDs00And20ToE0"/>
  </TABLE-ROW>

  ...

  <TABLE-ROW ID="ES_OBDIICommonServices.TAB_PIDsSupported.TABROW_OBDSupportedIDE0">
    <SHORT-NAME>ODXSupportedIDE0</SHORT-NAME>
    <LONG-NAME>ODX Supported ID E0</LONG-NAME>
    <KEY>E0</KEY>
    <STRUCTURE-REF ID-REF="STRUC_OBDSupportedIDsOfPIDs00And20ToE0"/>
  </TABLE-ROW>
```

</TABLE>

The structure of the result of the above REQUEST is described with the following POS-RESPONSE.

```
<POS-RESPONSE ID="Resp_Service01RequestCurrentPowertrainDiagnosticData">
  <SHORT-NAME>Service01RequestCurrentPowertrainDiagnosticData</SHORT-NAME>
  <LONG-NAME>Service 0x01 - Request current powertrain diagnostic data</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="CODED-CONST">
      <SHORT-NAME>RequestCurrentPowertrainDiagnosticDataResponseSID</SHORT-NAME>
      <LONG-NAME>Request current powertrain diagnostic data response SID</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <CODED-VALUE>65</CODED-VALUE>
      <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
        <BIT-LENGTH>8</BIT-LENGTH>
      </DIAG-CODED-TYPE>
    </PARAM>
    <PARAM xsi:type="VALUE">
      <SHORT-NAME>PIDsSupported</SHORT-NAME>
      <LONG-NAME>PIDs supported</LONG-NAME>
      <BYTE-POSITION>1</BYTE-POSITION>
      <DOP-REF ID-REF="EOPDUF_PIDsSupportedServices01060809Response"/>
    </PARAM>
  </PARAMS>
</POS-RESPONSE>
```

The resulting PIDs are defined in the TABLE “PIDsSupported”. The reference to this TABLE is established via the PARAM “PIDsSupported”.

The STRUCTURE “OBD supported IDs of PIDs 0x00 and 0x20 to 0xE0” is used for the interpretation which PIDs inside a PID-Range (00, 20, ... E0) are supported.

```
<STRUCTURE ID="STRUC_OBDSupportedIDsOfPIDs00And20ToE0">
  <SHORT-NAME>OBDSupportedIDsOfPIDs00And20ToE0</SHORT-NAME>
  <LONG-NAME>OBD supported IDs of PIDs 0x00 and 0x20 to 0xE0</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="VALUE">
      <SHORT-NAME>OBDSupportedID01</SHORT-NAME>
      <LONG-NAME>OBD Supported ID 01</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <BIT-POSITION>7</BIT-POSITION>
      <DOP-REF ID-REF="DOP_NotSupportedSupported"/>
    </PARAM>
    <PARAM xsi:type="VALUE">
      <SHORT-NAME>OBDSupportedID02</SHORT-NAME>
      <LONG-NAME>OBD Supported ID 02</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <BIT-POSITION>6</BIT-POSITION>
      <DOP-REF ID-REF="DOP_NotSupportedSupported"/>
    </PARAM>
    ...
    <PARAM xsi:type="VALUE">
      <SHORT-NAME>OBDSupportedID1F</SHORT-NAME>
      <LONG-NAME>OBD Supported ID 1F</LONG-NAME>
      <BYTE-POSITION>3</BYTE-POSITION>
      <BIT-POSITION>1</BIT-POSITION>
      <DOP-REF ID-REF="DOP_NotSupportedSupported"/>
    </PARAM>
    <PARAM xsi:type="VALUE">
      <SHORT-NAME>OBDSupportedID20</SHORT-NAME>
```

```

    <LONG-NAME>OBD Supported ID 20</LONG-NAME>
    <BYTE-POSITION>3</BYTE-POSITION>
    <BIT-POSITION>0</BIT-POSITION>
    <DOP-REF ID-REF="DOP_NotSupportedSupported"/>
  </PARAM>
  ...
</PARAMS>
</STRUCTURE>

```

7.6.2.4 ISO°15031-5 — Request multiple PIDs from vehicle

The request and response message described in this section can be used within ISO 15765-4.

The external test equipment requests a combination of a maximum of six (6) PIDs in one request message to gain best performance of displaying current data.

- PID 0x15: Bank 1 - Sensor 2, PID is supported by ECU #1;
- PID 0x01: Number of emission-related DTCs and MIL status, PID is supported by ECU #1 and #2;
- PID 0x05: Engine coolant temperature, PID is supported by ECU #1;
- PID 0x03: Fuel system 1 status, PID is supported by ECU #1;
- PID 0x0C: Engine speed PID is supported by ECU #1; and
- PID 0x0D: Vehicle speed PID is supported by ECU #2.

Table 16 — Request current powertrain diagnostic data request message

Message direction:		External test equipment → All ECUs	
Message Type:		Request	
Data Byte	Description (All PID values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Request current powertrain diagnostic data request SID	01	SIDRQ
#2	PID: Bank 1 - Sensor 2	15	PID(15)
#3	PID: Number of emission-related DTCs and MIL status	01	PID(01)
#4	PID: Engine coolant temperature	05	PID(05)
#5	PID: Fuel system 1 status	03	PID(03)
#6	PID: Engine speed	0C	PID(0C)
#7	PID: Vehicle speed	0D	PID(0D)

Table 17 — ECU#1 response: Request current powertrain diagnostic data response message

Message direction:		ECU#1 → External test equipment	
Message Type:		Response	
Data Byte	Description (All PID values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Request current powertrain diagnostic data response SID	41	SIDPR
#2	PID: Engine coolant temperature	05	PID(05)
#3	Data byte A	6E	DATA(A)
#4	PID: Number of emission-related DTCs and MIL status	01	PID(01)
#5	MIL: ON; Number of emission-related DTCs: 03	83	DATA(A)
#6	Misfire -, Fuel system -, Comprehensive monitoring	33	DATA(B)
#7	Catalyst -, Heated catalyst -, ..., monitoring supported	FF	DATA(C)
#8	Catalyst -, Heated catalyst -, ..., monitoring test complete/not complete	63	DATA(D)
#9	PID: Bank 1 - Sensor 2	15	PID(15)
#10	Bank 2 - Sensor 2: 0,8 Volt	A0	DATA(A)
#11	Bank 2 - Sensor 2: 93,7 %	78	DATA(B)
#12	PID: Engine speed	0C	PID(0C)
#13	Data byte A: 667 rpm	0A	DATA(A)
#14	Data byte B: 667 rpm	6B	DATA(B)
#15	PID: Fuel system 1 status	03	PID(03)
#16	Data byte A: Closed loop - using oxygen sensor(s) as feedback for fuel control	02	DATA(A)
#17	Data byte B	00	DATA(B)

Table 18 — ECU#2 response: Request current powertrain diagnostic data response message

Message direction:		ECU#2 → External test equipment	
Message Type:		Response	
Data Byte	Description (All PID values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Request current powertrain diagnostic data response SID	41	SIDPR
#2	PID: Vehicle speed	0D	PID(0D)
#3	Data byte A	23	DATA(A)
#4	PID: Number of emission-related DTCs and MIL status	01	PID(01)
#5	MIL: OFF; Number of emission-related DTCs: 01	01	DATA(A)
#6	Comprehensive monitoring: supported, test complete	44	DATA(B)
#7	Catalyst -, Heated catalyst -, ..., monitoring supported	00	DATA(C)
#8	Catalyst -, Heated catalyst -, ..., monitoring test complete/not complete	00	DATA(D)

7.6.2.5 ISO°22901-2 - Request multiple PIDs from vehicle ODX sample

This case is handled with the use of SUB-COMPONENTS described in chapter 7.4.9.

If, for example, PID 0x13 is supported the following service should be executed to request PIDs from the vehicle. The relationship between its request, response, table and table rows is defined as in clause 7.6.2.3.

```
<DIAG-SERVICE ID="DiagnServi_Service01RequestCurrentPowertrainDiagnosticDataPID13"
ADDRESSING="FUNCTIONAL" >
  <SHORT-NAME>Service01RequestCurrentPowertrainDiagnosticDataPID13</SHORT-NAME>
  <LONG-NAME>Service $01 - Request current powertrain diagnostic data (PID $13)</LONG-
NAME>
  <REQUEST-REF ID-REF="Req_Service01RequestCurrentPowertrainDiagnosticDataPID13" />
  <POS-RESPONSE-REFS>
    <POS-RESPONSE-REF ID-
REF="Resp_Service01RequestCurrentPowertrainDiagnosticDataPID13" />
  </POS-RESPONSE-REFS>
</DIAG-SERVICE>
```

Examples of the STRUCTUREs reference by the TABLE-ROW elements can be found in clause 7.6.9.

7.6.3 Service 0x02 — Request powertrain freeze frame data

7.6.3.1 ISO°15031-5 — Request powertrain freeze frame with PID 0x02

Now the external test equipment requests PID 0x02 of freeze frame 0x00 from the vehicle. Since the ECU#2 (TCM) doesn't store a freeze frame data record, only the ECU#1 (ECM) will send a response message. In this example, the freeze frame data are stored based on a DTC P0130 occurrence. The parameter value of PID 0x02 "DTC that caused required freeze frame data storage" is set to the DTC P0130.

Table 19 — Request powertrain freeze frame data request message

Message Direction:		External test equipment → All ECUs	
Message Type:		Request	
Data Byte	Description (all values are in hexadecimal)	Byte Value	Mnemonic
#1	Request powertrain freeze frame data request SID	0x02	SIDRQ
#2	PID: DTC that caused required freeze frame data storage	0x02	PID
#3	Frame #	0x00	FRNO

Table 20 — Request powertrain freeze frame data response message

Message Direction:		ECU#1 → External test equipment	
Message Type:		Response	
Data Byte	Description (all values are in hexadecimal)	Byte Value	Mnemonic
#1	Request powertrain freeze frame data response SID	0x42	SIDRQ
#2	PID: DTC that caused required freeze frame data storage	0x02	PID
#3	Frame #	0x00	FRNO
#4	DTC High Byte of P0130	0x01	DATA_A
#5	DTC Low Byte of P0130	0x30	DATA_B

NOTE: ECU#2 does not store freeze frame data and therefore does not send a response message.

7.6.3.2 ISO 22900-2 — Request powertrain freeze frame PID 0x02 in ODX sample

```
<DIAG-SERVICE ID="DiagnServi_Service02RequestPowertrainFreezeFrameData"
  ADDRESSING="FUNCTIONAL">
  <SHORT-NAME>Service02RequestPowertrainFreezeFrameData</SHORT-NAME>
  <LONG-NAME>Service 0x02 - Request powertrain freeze frame data</LONG-NAME>
  <REQUEST-REF ID-REF="Req_Service02RequestPowertrainFreezeFrameData" />
  <POS-RESPONSE-REFS>
    <POS-RESPONSE-REF ID-REF="Resp_Service02RequestPowertrainFreezeFrameData" />
  </POS-RESPONSE-REFS>
</DIAG-SERVICE>

<REQUEST ID="Req_Service02RequestPowertrainFreezeFrameData">
  <SHORT-NAME>Service02RequestPowertrainFreezeFrameData</SHORT-NAME>
  <LONG-NAME>Service 0x02 - Request powertrain freeze frame data</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="CODED-CONST">
      <SHORT-NAME>RequestPowertrainFreezeFrameDataRequestSID</SHORT-NAME>
      <LONG-NAME>Request powertrain freeze frame data request SID</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <CODED-VALUE>2</CODED-VALUE>
      <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
        <BIT-LENGTH>8</BIT-LENGTH>
      </DIAG-CODED-TYPE>
    </PARAM>
    ... <!-- field with PIDs and frame numbers -->
  </PARAMS>
</REQUEST>

<POS-RESPONSE ID="Resp_Service02RequestPowertrainFreezeFrameData">
  <SHORT-NAME>Service02RequestPowertrainFreezeFrameData</SHORT-NAME>
  <LONG-NAME>Service 0x02 - Request powertrain freeze frame data</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="CODED-CONST">
      <SHORT-NAME>RequestPowertrainFreezeFrameDataResponseSID</SHORT-NAME>
      <LONG-NAME>Request powertrain freeze frame data response SID</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <CODED-VALUE>66</CODED-VALUE>
      <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
        <BIT-LENGTH>8</BIT-LENGTH>
      </DIAG-CODED-TYPE>
    </PARAM>
    <PARAM>
      <!-- Get the freeze frame data -->
    </PARAM>
  </PARAMS>
</POS-RESPONSE>
```

7.6.3.3 ISO 15031-5 — Request powertrain freeze frame with multiple PIDs

The external test equipment requests the parameter value of PID 0x0C “Engine Speed”, PID 0x05 “Engine coolant temperature”, and PID 0x04 “Load” stored in the freeze frame.

Table 21 — Request powertrain freeze frame data request message

Message Direction:		External test equipment → All ECUs	
Message Type:		Request	
Data Byte	Description (all values are in hexadecimal)	Byte Value	Mnemonic
#1	Request powertrain freeze frame data request SID	0x02	SIDRQ
#2	PID: Engine Speed	0x0C	PID
#3	Frame #	0x00	FRNO
#4	PID: Engine coolant temperature	0x05	PID
#5	Frame #	0x00	FRNO
#4	PID: Load	0x04	PID
#5	Frame #	0x00	FRNO

Table 22 — Request powertrain freeze frame data response message

Message Direction:		ECU#1 → External test equipment	
Message Type:		Response	
Data Byte	Description (all values are in hexadecimal)	Byte Value	Mnemonic
#1	Request powertrain freeze frame data response SID	0x42	SIDRQ
#2	PID: Engine Speed	0x0C	PID
#3	Frame #	0x00	FRNO
#4	High Byte: Engine Speed: 2080 rpm	0x20	DATA_A
#5	Low Byte: Engine Speed: 2080 rpm	0x80	DATA_B
#6	PID: Load	0x04	PID
#7	Frame #	0x00	FRNO
#8	Load: 50.2 %	0x80	DATA_A
#9	PID: Engine coolant temperature	0x05	PID
#10	Frame #	0x00	FRNO
#11	Engine coolant temperature: 0 °C	0x28	DATA_A

7.6.3.4 ISO 22900-2 — Request powertrain freeze frame with multiple PIDs in ODX sample

Requesting a freeze frame for a single PID or multiple PIDs does not impact the ODX data. Therefore, the examples shown are identical to clause 7.6.3.2.

```

<DIAG-SERVICE ID="DiagnServi_Service02RequestPowertrainFreezeFrameData"
  ADDRESSING="FUNCTIONAL">
  <SHORT-NAME>Service02RequestPowertrainFreezeFrameData</SHORT-NAME>
  <LONG-NAME>Service 0x02 - Request powertrain freeze frame data</LONG-NAME>
  <REQUEST-REF ID-REF="Req_Service02RequestPowertrainFreezeFrameData"/>
  <POS-RESPONSE-REFS>
    <POS-RESPONSE-REF ID-REF="Resp_Service02RequestPowertrainFreezeFrameData"/>
  </POS-RESPONSE-REFS>
</DIAG-SERVICE>

```

```

<REQUEST ID="Req_Service02RequestPowertrainFreezeFrameData">
  <SHORT-NAME>Service02RequestPowertrainFreezeFrameData</SHORT-NAME>
  <LONG-NAME>Service 0x02 - Request powertrain freeze frame data</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="CODED-CONST">
      <SHORT-NAME>RequestPowertrainFreezeFrameDataRequestSID</SHORT-NAME>
      <LONG-NAME>Request powertrain freeze frame data request SID</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <CODED-VALUE>2</CODED-VALUE>
      <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
        <BIT-LENGTH>8</BIT-LENGTH>
      </DIAG-CODED-TYPE>
    </PARAM>
    ... <!-- field with PIDs and frame numbers -->
  </PARAMS>
</REQUEST>

<POS-RESPONSE ID="Resp_Service02RequestPowertrainFreezeFrameData">
  <SHORT-NAME>Service02RequestPowertrainFreezeFrameData</SHORT-NAME>
  <LONG-NAME>Service 0x02 - Request powertrain freeze frame data</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="CODED-CONST">
      <SHORT-NAME>RequestPowertrainFreezeFrameDataResponseSID</SHORT-NAME>
      <LONG-NAME>Request powertrain freeze frame data response SID</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <CODED-VALUE>66</CODED-VALUE>
      <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
        <BIT-LENGTH>8</BIT-LENGTH>
      </DIAG-CODED-TYPE>
    </PARAM>
    <PARAM>
      <!-- Get the freeze frame data -->
    </PARAM>
  </PARAMS>
</POS-RESPONSE>

```

7.6.4 Service 0x03, 0x07 and 0x0A — Request emissions-related DTC

7.6.4.1 ISO°22900-2 — DIAG-SERVICE 0x03 with LONG-NAME and SHORT-NAME sample

```

<DIAG-SERVICE ID="DiagnServi_Service03RequestEmissionRelatedDiagnosticTroubleCodes"
  ADDRESSING="FUNCTIONAL">
  <SHORT-NAME>Service03RequestEmissionRelatedDiagnosticTroubleCodes</SHORT-NAME>
  <LONG-NAME>Service 0x03 - Request emission-related diagnostic trouble codes</LONG-NAME>
  <REQUEST-REF ID-REF="Req_Service03RequestEmissionRelatedDiagnosticTroubleCodes" />
  <POS-RESPONSE-REFS>
    <POS-RESPONSE-REF
      ID-REF="Resp_Service03RequestEmissionRelatedDiagnosticTroubleCodes" />
    </POS-RESPONSE-REFS>
</DIAG-SERVICE>

```

7.6.4.2 ISO°15031-5 service 0x03, 0x07, 0x0A — Request emission-related DTCs example

The request and response message described in this section can be used within ISO 15765-4.

The example below shows how the “Request emission-related DTCs” service shall be implemented. The external test equipment requests emission-related DTCs from the vehicle. The ECU#1 (ECM) has six (6) DTCs stored, the ECU #2 (TCM) has one (1) DTC stored, and the ECU #3 (ABS/Traction Control) has no DTC stored.

- ECU #1 (ECM): P0143, P0196, P0234, P02CD, P0357, P0A24
- ECU #2 (TCM): P0443
- ECU #3 (ABS/Traction Control): no emission-related DTC stored

Table 23 — Request emission-related diagnostic trouble codes request message

Message direction:		External test equipment → All ECUs	
Message Type:		Request	
Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Request emission-related DTCs request SID	03/07/0A	SIDRQ

Table 24 — Request emission-related diagnostic trouble codes response message

Message direction:		ECU #1 → External test equipment	
Message Type:		Response	
Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Request emission-related DTCs response SID	43/47/4A	SIDPR
#2	# of DTC {number of emission-related DTCs stored in this ECU}	06	#OFDTC
#3	DTC High Byte of P0143	01	DTC1HI
#4	DTC Low Byte of P0143	43	DTC1LO
#5	DTC High Byte of P0196	01	DTC2HI
#6	DTC Low Byte of P0196	96	DTC2LO
#7	DTC High Byte of P0234	02	DTC3HI
#8	DTC Low Byte of P0234	34	DTC3LO
#9	DTC High Byte of P02CD	02	DTC4HI
#10	DTC Low Byte of P02CD	CD	DTC4LO
#11	DTC High Byte of P0357	03	DTC5HI
#12	DTC Low Byte of P0357	57	DTC5LO
#13	DTC High Byte of P0A24	0A	DTC6HI
#14	DTC Low Byte of P0A24	24	DTC6LO

Table 25 — Request emission-related diagnostic trouble codes response message

Message direction:		ECU #3 → External test equipment	
Message Type:		Response	
Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Request emission-related DTCs response SID	43/47/4A	SIDPR
#2	# of DTC {number of emission-related DTCs stored in this ECU}	00	#OFDTC

Table 26 — Request emission-related diagnostic trouble codes response message

Message direction:		ECU #2 → External test equipment	
Message Type:		Response	
Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Request emission-related DTCs response SID	43/47/4A	SIDPR
#2	# of DTC {number of emission-related DTCs stored in this ECU}	01	#OFDTC
#3	DTC High Byte of P0443	04	DTC1HI
#4	DTC Low Byte of P0443	43	DTC1LO

7.6.4.3 ISO°22901-2 — Request emission-related DTCs in ODX sample

```

<REQUEST ID="Req_Service03RequestEmissionRelatedDiagnosticTroubleCodes">
  <SHORT-NAME>Service03RequestEmissionRelatedDiagnosticTroubleCodes</SHORT-NAME>
  <LONG-NAME>Service 0x03 - Request emission-related
                                diagnostic trouble codes</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="CODED-CONST">
      <SHORT-NAME>RequestEmissionRelatedDiagnostic
                                TroubleCodesRequestSID</SHORT-NAME>
      <LONG-NAME>Request emission-related diagnostic
                                trouble codes request SID</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <CODED-VALUE>3</CODED-VALUE>
      <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32"
                                xsi:type="STANDARD-LENGTH-TYPE">
        <BIT-LENGTH>8</BIT-LENGTH>
      </DIAG-CODED-TYPE>
    </PARAM>
  </PARAMS>
</REQUEST>

<POS-RESPONSE ID="Resp_Service03RequestEmissionRelatedDiagnosticTroubleCodes">
  <SHORT-NAME>Service03RequestEmissionRelatedDiagnosticTroubleCodes</SHORT-NAME>
  <LONG-NAME>Service 0x03 - Request emission-related diagnostic trouble codes</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="CODED-CONST">
      <SHORT-NAME>RequestEmissionRelatedDiagnosticTrouble
                                CodesResponseSID</SHORT-NAME>
      <LONG-NAME>Request emission-related diagnostic trouble
                                codes response SID</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <CODED-VALUE>67</CODED-VALUE>
      <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
        <BIT-LENGTH>8</BIT-LENGTH>
      </DIAG-CODED-TYPE>
    </PARAM>
    <PARAM SEMANTIC="DATA" xsi:type="VALUE">
      <SHORT-NAME>List_of_DTC</SHORT-NAME>
      <LONG-NAME>List of DTC</LONG-NAME>
      <BYTE-POSITION>1</BYTE-POSITION>
      <DOP-REF ID-REF="List_of_DTC"/>
    </PARAM>
  </PARAMS>
</POS-RESPONSE>

<STRUCTURE ID="STRUC_ListOfDTC">
  <SHORT-NAME>ListOfDTC</SHORT-NAME>
  <LONG-NAME>List of DTC</LONG-NAME>
  <PARAMS>
    <PARAM SEMANTIC="DATA" xsi:type="VALUE">

```

```

        <SHORT-NAME>TestDataObjectQual</SHORT-NAME>
        <LONG-NAME>TestDataObjectName</LONG-NAME>
        <BYTE-POSITION>0</BYTE-POSITION>
        <DOP-REF ID-REF="DTC-DOP_RecordDataType" />
    </PARAM>
</PARAMS>
</STRUCTURE>

<DTC-DOP ID="DTC-DOP_RecordDataType">
    <SHORT-NAME>RecordDataType</SHORT-NAME>
    <LONG-NAME>RecordDataType</LONG-NAME>
    <DIAG-CODED-TYPE BASE-TYPE-ENCODING="NONE" BASE-DATA-TYPE="A_UINT32"
        xsi:type="STANDARD-LENGTH-TYPE">
        <BIT-LENGTH>16</BIT-LENGTH>
    </DIAG-CODED-TYPE>
    <PHYSICAL-TYPE BASE-DATA-TYPE="A_UINT32" DISPLAY-RADIX="HEX" />
    <COMPU-METHOD>
        <CATEGORY>IDENTICAL</CATEGORY>
    </COMPU-METHOD>
    <DTCS>
        <DTC ID="_OBD_ECU_78">
            <SHORT-NAME>DTC08</SHORT-NAME>
            <TROUBLE-CODE>8</TROUBLE-CODE>
            <TEXT>Engine Position System Performance</TEXT>
        </DTC>
    </DTCS>
</DTC-DOP>

```

7.6.5 Service 0x04 — Clear/Reset emission-related diagnostic information

7.6.5.1 ISO°15031-5 — Clear/Reset Emission-Related Diagnostic Information example

The example below shows how the “Clear/reset emission-related diagnostic information” service shall be implemented if ignition is ON and with the engine not running.

The external test equipment commands the vehicle to “Clear/reset emission-related diagnostic information”.

Table 27 — Clear/reset emission-related diagnostic information request message

Message Direction:		External test equipment → All ECUs	
Message Type:		Request	
Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Clear/reset emission-related diagnostic information request SID	0x04	SIDRQ

Table 28 — Clear/reset emission-related diagnostic information response message

Message Direction:		ECU#1 → External test equipment	
Message Type:		Response	
Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Clear/reset emission-related diagnostic information response SID	0x44	SIDPR

Table 29 — Clear/reset emission-related diagnostic information response message

Message Direction:		ECU#2 → External test equipment	
Message Type:		Response	
Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Clear/reset emission-related diagnostic information response SID	0x44	SIDPR

Table 30 — Negative response message

Message Direction:		ECU#3 → External test equipment	
Message Type:		Response	
Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Negative Response Service Identifier	0x7F	SIDNR
#2	Clear/reset emission-related diagnostic information request SID	0x04	SIDRQ
#3	Negative Response Code: conditionsNotCorrect	0x22	NR_CNC

7.6.5.2 ISO°22901-2 — Clear/Reset Emission-Related Diagnostic Information in ODX sample

```

<DIAG-SERVICE ID="DiagnServi_Service04ClearResetEmissionRelatedDiagnosticInformation"
  ADDRESSING="FUNCTIONAL">
  <SHORT-NAME>Service04ClearResetEmissionRelatedDiagnosticInformation</SHORT-NAME>
  <LONG-NAME>Service 0x04 - Clear/reset emission-related
    diagnostic information</LONG-NAME>
  <REQUEST-REF ID-REF="Req_Service04ClearResetEmissionRelatedDiagnosticInformation"/>
  <POS-RESPONSE-REFS>
    <POS-RESPONSE-REF ID-REF="Resp_Service04ClearResetEmissionRelated
      DiagnosticInformation"/>
  </POS-RESPONSE-REFS>
</DIAG-SERVICE>

<REQUEST ID="Req_Service04ClearResetEmissionRelatedDiagnosticInformation">
  <SHORT-NAME>Service04ClearResetEmissionRelatedDiagnosticInformation</SHORT-NAME>
  <LONG-NAME>Service 0x04 - Clear/reset emission-related
    diagnostic information</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="CODED-CONST">
      <SHORT-NAME>ClearResetEmissionRelated
        DiagnosticInformationRequestSID</SHORT-NAME>
      <LONG-NAME>Clear/reset emission-related
        diagnostic information request SID</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <CODED-VALUE>3</CODED-VALUE>
      <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
        <BIT-LENGTH>8</BIT-LENGTH>
      </DIAG-CODED-TYPE>
    </PARAM>
  </PARAMS>
</REQUEST>

<POS-RESPONSE ID="Resp_Service04ClearResetEmissionRelatedDiagnosticInformation">
  <SHORT-NAME>Service04ClearResetEmissionRelatedDiagnosticInformation</SHORT-NAME>
  <LONG-NAME>Service 0x04 - Clear/reset emission-related
    diagnostic information</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="CODED-CONST">
      <SHORT-NAME>ClearResetEmissionRelated
        DiagnosticInformationResponseSID</SHORT-NAME>

```

```

<LONG-NAME>Clear/reset emission-related
                        diagnostic information response SID</LONG-NAME>
<BYTE-POSITION>0</BYTE-POSITION>
<CODED-VALUE>68</CODED-VALUE>
<DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
    <BIT-LENGTH>8</BIT-LENGTH>
</DIAG-CODED-TYPE>
</PARAM>
</PARAMS>
</POS-RESPONSE>

```

7.6.6 Service 0x06 — Request on-board monitoring test results

7.6.6.1 ISO^o15031-5 — Request on-board monitoring test results example

The request and response message described in this section can be used within ISO 15765-4.

The external test equipment sends a “Request on-board monitoring test results for specific monitored systems” message with one supported OBDMID in the request message to the vehicle. In this example, the request message includes the following OBDMID:

— request message: OBDMID 0x01 - Oxygen Sensor Monitor Bank 1 - Sensor 1

Table 31 — Request oxygen sensor monitoring test results request message

Message direction:		External test equipment → All ECUs	
Message Type:		Request	
Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Request on-board monitoring test results for specific monitored systems request SID	06	SIDRQ
#2	OBDMID: 01 - Oxygen Sensor Monitor Bank 1 - Sensor 1	01	OBDMID

Table 32 — Request oxygen sensor monitoring test results response message

Message direction:		ECU #1 → External test equipment	
Message Type:		Response	
Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Request on-board monitoring test results for specific monitored systems response SID	46	SIDPRQ
#2	OBDMID: 01 - Oxygen Sensor Monitor Bank 1 - Sensor 1	01	OBDMID
#3	Standardized Test ID: 01 - Rich to lean sensor threshold voltage (constant)	01	STID
#4	Unit And Scaling ID: Voltage	0A	UASID
#5	Test Value High Byte:	0B	TESTVAL
#6	Test Value Low Byte: 0,365 V	B0	TESTVAL
#7	Minimum Test Limit High Byte:	0B	MINLIMIT
#8	Minimum Test Limit Low Byte: 0,365 V	B0	MINLIMIT
#9	Maximum Test Limit High Byte:	0B	MAXLIMIT
#10	Maximum Test Limit Low Byte: 0,365 V	B0	MAXLIMIT
#11	OBDMID: 01 - Oxygen Sensor Monitor Bank 1 - Sensor 1	01	OBDMID
#12	Standardized Test ID: 05 - Rich to lean sensor switch time (calculated)	05	STID
#13	Unit And Scaling ID: Time	10	UASID
#14	Test Value High Byte	00	TESTVAL
#15	Test Value Low Byte: 0,072 s (0 min, 0 s)	48	TESTVAL
#16	Minimum Test Limit High Byte	00	MINLIMIT
#17	Minimum Test Limit Low Byte: 0,000 s (0 min, 0 s)	00	MINLIMIT
#18	Maximum Test Limit High Byte	00	MAXLIMIT
#19	Maximum Test Limit Low Byte: 0,100 s (0 min, 0 s)	64	MAXLIMIT
#20	OBDMID: 01 - Oxygen Sensor Monitor Bank 1 - Sensor 1	01	OBDMID
#21	Manufacturer Defined Test ID: 133 (The name of this Test ID shall be documented in the vehicle Service Information.)	85	MDTID
#22	Unit And Scaling ID: Counts	24	UASID
#23	Test Value High Byte	00	TESTVAL
#24	Test Value Low Byte: 150 counts	96	TESTVAL
#25	Minimum Test Limit High Byte	00	MINLIMIT
#26	Minimum Test Limit Low Byte: 75 counts	4B	MINLIMIT
#27	Maximum Test Limit High Byte	FF	MAXLIMIT
#28	Maximum Test Limit Low Byte: 65535 counts	FF	MAXLIMIT

NOTE ECU#2 does not support any Test IDs and therefore does not send a response message.

7.6.6.2 ISO 22900-2 — Request on-board monitoring test results in ODX sample

```

<DIAG-SERVICE ID="DiagnServi_Service06RequestOnBoardMonitoringTestResults
    ForSpecificMonitoredSystems" ADDRESSING="FUNCTIONAL">
  <SHORT-NAME>Service06RequestOnBoardMonitoringTestResults
    ForSpecificMonitoredSystems</SHORT-NAME>
  <LONG-NAME>Service 0x06 - Request on-board monitoring test results
    for specific monitored systems</LONG-NAME>
  <REQUEST-REF ID-REF="Req_Service06RequestOnBoardMonitoringTestResults

```



```

ForSpecificMonitoredSystems" />
<POS-RESPONSE-REFS>
  <POS-RESPONSE-REF ID-REF="Resp_Service06RequestOnBoardMonitoringTestResults
ForSpecificMonitoredSystems" />
</POS-RESPONSE-REFS>
</DIAG-SERVICE>

```

The following Request is used to determine which OBD-MIDs are supported by the ECU.

```

<REQUEST ID="Req_Service06RequestOnBoardMonitoringTestResultsForSpecificMonitoredSystems">
  <SHORT-NAME>Service06RequestOnBoardMonitoringTestResults
ForSpecificMonitoredSystems</SHORT-NAME>
  <LONG-NAME>Service 0x06 - Request on-board monitoring test results
for specific monitored systems</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="CODED-CONST">
      <SHORT-NAME>RequestOnBoardMonitoringTestResults
ForSpecificMonitoredSystemsRequestSID</SHORT-NAME>
      <LONG-NAME>Request on-board monitoring test results
for specific monitored systems request SID</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <CODED-VALUE>6</CODED-VALUE>
      <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
        <BIT-LENGTH>8</BIT-LENGTH>
      </DIAG-CODED-TYPE>
    </PARAM>
    <PARAM SEMANTIC="ID" ID="Param_OBDMID" xsi:type="TABLE-KEY">
      <SHORT-NAME>OBDMID</SHORT-NAME>
      <LONG-NAME>OBDMID</LONG-NAME>
      <BYTE-POSITION>1</BYTE-POSITION>
      <TABLE-REF ID-REF="ES_OBDIICommonServices.
TAB_OnboardMonitoringTestResultsReadSupportedOBDMIDs" />
    </PARAM>
  </PARAMS>
</REQUEST>

<TABLE ID="ES_OBDIICommonServices.
TAB_OnboardMonitoringTestResultsReadSupportedOBDMIDs" SEMANTIC="OBDMID">
  <SHORT-NAME>OnboardMonitoringTestResultsReadSupportedOBDMIDs</SHORT-NAME>
  <LONG-NAME>Onboard Monitoring Test Results Read supported OBDMIDs</LONG-NAME>
  <KEY-DOP-REF ID-REF="ES_OBDIICommonServices.DOP_OBDMIDsSupported" />
  <TABLE-ROW ID="ES_OBDIICommonServices.
TAB_OnboardMonitoringTestResultsReadSupportedOBDMIDs.
TABROW_OBDMID00OBDMIDSupported01To20">
    <SHORT-NAME>OBDMID00OBDMIDSupported01To20</SHORT-NAME>
    <LONG-NAME>(OBDMID 0x00) OBDMID supported, 0x01 to 0x20</LONG-NAME>
    <KEY>0</KEY>
    <STRUCTURE-REF ID-REF="STRUC_ReadSupportedOBDMIDsOBDMID00OBDMIDSupported01To20" />
  </TABLE-ROW>
</TABLE>

```

The Structure described below is used to interpret which OBDMIDs are supported.

```

<STRUCTURE ID="STRUC_ReadSupportedOBDMIDsOBDMID00OBDMIDSupported01To20">
  <SHORT-NAME>ReadSupportedOBDMIDsOBDMID00OBDMIDSupported01To20</SHORT-NAME>
  <LONG-NAME>Read supported OBDMIDs (OBDMID 0x00) OBDMID supported, 0x01to 0x20</LONG-NAME>
  <PARAMS>
    <PARAM SEMANTIC="DATA" xsi:type="VALUE">
      <SHORT-NAME>SupportedOBDMIDs01To08</SHORT-NAME>
      <LONG-NAME>Supported OBDMIDs 0x01 to 0x08</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <DOP-REF ID-REF="STRUC_SupportedOBDMIDs01To08" />
    </PARAM>
  </PARAMS>
</STRUCTURE>

```

```

...
</PARAMS>
</STRUCTURE>

<STRUCTURE ID="STRUC_SupportedOBDMIDs01To08">
  <SHORT-NAME>SupportedOBDMIDs01To08</SHORT-NAME>
  <LONG-NAME>Supported OBDMIDs 0x01 to 0x08</LONG-NAME>
  <BYTE-SIZE>1</BYTE-SIZE>
  <PARAMS>
    <PARAM SEMANTIC="DATA" xsi:type="VALUE">
      <SHORT-NAME>OBDMID0x08Supported</SHORT-NAME>
      <LONG-NAME>OBDMID 0x08 supported</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <DOP-REF ID-REF="ES_OBDIICommonServices.DOP_OBDMIDsSupported"/>
    </PARAM>
  </PARAMS>
</STRUCTURE>

```

To request the Data stored for a specific OBDMID, the following TABLE is used.

```

<TABLE ID="ES_OBDIICommonServices.TAB_OnBoardMonitoringTestResultsReadOBDMIDData"
  SEMANTIC="OBDMID">
  <SHORT-NAME>OnBoardMonitoringTestResultsReadOBDMIDData</SHORT-NAME>
  <LONG-NAME>Onboard Monitoring Test Results Read OBDMID Data</LONG-NAME>
  <KEY-DOP-REF ID-REF="ES_OBDIICommonServices.DOP_OBDMIDsSupported"/>
  <TABLE-ROW ID="ES_OBDIICommonServices.
    TAB_OnBoardMonitoringTestResultsReadOBDMIDData.
    TABROW_OBDMID04OxygenSensorMonitorBank1ToSensor4">
    <SHORT-NAME>OBDMID04OxygenSensorMonitorBank1ToSensor4</SHORT-NAME>
    <LONG-NAME>(OBDMID 0x04) Oxygen Sensor Monitor Bank 1 to Sensor 4</LONG-NAME>
    <KEY>4</KEY>
    <STRUCTURE-REF ID-REF="STRUC_ReadOBDMIDDataPROBDMID04OxygenSensor
      MonitorBank1ToSensor4"/>
  </TABLE-ROW>
</TABLE>

<STRUCTURE ID="STRUC_ReadOBDMIDDataPROBDMID04OxygenSensorMonitorBank1ToSensor4">
  <SHORT-NAME>ReadOBDMIDDataPROBDMID04OxygenSensorMonitorBank1ToSensor4</SHORT-NAME>
  <LONG-NAME>Read OBDMID Data PR (OBDMID 0x04) Oxygen Sensor
    Monitor Bank 1 to Sensor 4</LONG-NAME>
  <PARAMS>
    <PARAM SEMANTIC="DATA" xsi:type="VALUE">
      <SHORT-NAME>OBDMID04OxygenSensorMonitorBank1Sensor4Data</SHORT-NAME>
      <LONG-NAME>OBDMID 04 Oxygen Sensor Monitor Bank 1 Sensor 4 Data</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <DOP-REF ID-REF="EOPDUF_DataRecord"/>
    </PARAM>
  </PARAMS>
</STRUCTURE>

```

For each OBDMID the Interpretation is determined using Unit-and-Scaling-ID. This is represented by an END-OF-PDU-FIELD and a MUX.

```

<END-OF-PDU-FIELD ID="EOPDUF_DataRecord">
  <SHORT-NAME>DataRecord</SHORT-NAME>
  <LONG-NAME>Data Record</LONG-NAME>
  <BASIC-STRUCTURE-REF ID-REF="STRUC_DataRecord"/>
</END-OF-PDU-FIELD>

<STRUCTURE ID="STRUC_DataRecord">
  <SHORT-NAME>DataRecord</SHORT-NAME>
  <LONG-NAME>Data Record</LONG-NAME>
  <PARAMS>

```

```

    <PARAM SEMANTIC="DATA" xsi:type="VALUE">
      <SHORT-NAME>OBDMID</SHORT-NAME>
      <LONG-NAME>OBDMID</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <DOP-REF ID-REF="_2"/>
    </PARAM>
    <PARAM SEMANTIC="DATA" xsi:type="VALUE">
      <SHORT-NAME>TID</SHORT-NAME>
      <LONG-NAME>TID</LONG-NAME>
      <BYTE-POSITION>1</BYTE-POSITION>
      <DOP-REF ID-REF="_169"/>
    </PARAM>
    <PARAM SEMANTIC="DATA" xsi:type="VALUE">
      <SHORT-NAME>UnitAndScalingID</SHORT-NAME>
      <LONG-NAME>Unit-And-Scaling-ID</LONG-NAME>
      <BYTE-POSITION>2</BYTE-POSITION>
      <DOP-REF ID-REF="_2"/>
    </PARAM>
    <PARAM SEMANTIC="DATA" xsi:type="VALUE">
      <SHORT-NAME>Multiplexer</SHORT-NAME>
      <LONG-NAME>Multiplexer</LONG-NAME>
      <BYTE-POSITION>2</BYTE-POSITION>
      <DOP-REF ID-REF="_254"/>
    </PARAM>
  </PARAMS>
</STRUCTURE>

<MUX ID="MUX_Multiplexer">
  <SHORT-NAME>Multiplexer</SHORT-NAME>
  <LONG-NAME>Multiplexer</LONG-NAME>
  <BYTE-POSITION>1</BYTE-POSITION>
  <SWITCH-KEY>
    <BYTE-POSITION>0</BYTE-POSITION>
    <DATA-OBJECT-PROP-REF ID-REF="_2"/>
  </SWITCH-KEY>
  <DEFAULT-CASE>
    <SHORT-NAME>Case_Default</SHORT-NAME>
    <STRUCTURE-REF ID-REF="_384"/>
  </DEFAULT-CASE>
  <CASES>
    <CASE>
      <SHORT-NAME>Case_0x1</SHORT-NAME>
      <STRUCTURE-REF ID-REF="_385"/>
      <LOWER-LIMIT>0x01</LOWER-LIMIT>
      <UPPER-LIMIT>0x01</UPPER-LIMIT>
    </CASE>
  </CASES>
</MUX>

<POS-RESPONSE ID="Resp_Service06RequestOnBoardMonitoringTestResults
  ForSpecificMonitoredSystems">
  <SHORT-NAME>Service06RequestOnBoardMonitoringTestResults
    ForSpecificMonitoredSystems</SHORT-NAME>
  <LONG-NAME>Service 0x06 - Request on-board monitoring test results
    for specific monitored systems</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="CODED-CONST">
      <SHORT-NAME>RequestOnBoardMonitoringTestResults
        ForSpecificMonitoredSystemsResponseSID</SHORT-NAME>
      <LONG-NAME>Request on-board monitoring test results
        for specific monitored systems response SID</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <CODED-VALUE>70</CODED-VALUE>
      <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
        <BIT-LENGTH>8</BIT-LENGTH>
      </DIAG-CODED-TYPE>
    </PARAM>
  </PARAMS>
</POS-RESPONSE>

```

```

</PARAM>
...
</PARAMS>
</POS-RESPONSE>

```

7.6.7 Service 0x08 — Request control of on-board device

7.6.7.1 ISO°15031-5 — Request control of on-board device example

The request and response message described in this section can be used within ISO°15765-4.

The external test equipment sends a “Request control of on-board device” message with one (1) supported Test ID 0x01 to the vehicle.

Table 33 — Request control of on-board device request message

Message direction:	External test equipment → All ECUs		
Message Type:	Request		
Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Request control of on-board device request SID	08	SIDRQ
#2	Test ID: 01 - Evaporative system leak test	01	TID

Table 34 — Request control of on-board device response message

Message direction:	ECU #1 → External test equipment		
Message Type:	Response		
Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Request control of on-board device response SID	48	SIDPR
#2	Test ID: 01 - Evaporative system leak test	01	TID

In the following example, the conditions of the system are not proper to run the Evaporative system leak test. Therefore, the ECM (ECU #1) responds with a negative response message with response code 0x22 - conditionsNotCorrect. The TCM (ECU #2) does not respond because it previously reported that it does not support the Evaporative system leak test.

Table 35 — Request control of on-board device request message

Message direction:	External test equipment → All ECUs		
Message Type:	Request		
Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Request control of on-board device request SID	08	SIDRQ
#2	Test ID: 01 - Evaporative system leak test	01	TID

Table 36 — Negative response message

Message direction:		ECU#1 → External test equipment	
Message Type:		Response	
Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Negative Response Service Identifier	7F	SIDNR
#2	Request control of on-board device request SID	08	SIDRQ
#3	Negative Response Code: conditionsNotCorrect	22	NR_CNC

7.6.7.2 ISO 22900-2 — Request control of on-board device in ODX sample

```

<DIAG-SERVICE ID="DiagnServi_Service08RequestControlOfOnBoardSystemTestOrComponentent "
  ADDRESSING="FUNCTIONAL">
  <SHORT-NAME>Service08RequestControlOfOnBoardSystemTestOrComponentent</SHORT-NAME>
  <LONG-NAME>Service 0x08 - Request control of on-board system, test or
    componentent</LONG-NAME>
  <REQUEST-REF ID-REF="Req_Service08RequestControlOfOnBoardSystemTestOrComponentent "/>
  <POS-RESPONSE-REFS>
    <POS-RESPONSE-REF ID-REF="Resp_Service08RequestControlOfOnBoardSystem
      TestOrComponentent "/>
  </POS-RESPONSE-REFS>
</DIAG-SERVICE>

```

The following request is used to determine the supported TIDs of the ECU.

```

<REQUEST ID="Req_Service08RequestControlOfOnBoardSystemTestOrComponentent ">
  <SHORT-NAME>Service08RequestControlOfOnBoardSystemTestOrComponentent</SHORT-NAME>
  <LONG-NAME>Service 0x08 - Request control of on-board system,
    test or componentent</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="CODED-CONST">
      <SHORT-NAME>RequestControlOfOnBoardSystem
        TestOrComponententRequestSID</SHORT-NAME>
      <LONG-NAME>Request control of on-board system,
        test or componentent request SID</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <CODED-VALUE>3</CODED-VALUE>
      <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
        <BIT-LENGTH>8</BIT-LENGTH>
      </DIAG-CODED-TYPE>
    </PARAM>
    <PARAM SEMANTIC="TID" ID="Param_TID" xsi:type="TABLE-KEY">
      <SHORT-NAME>TID</SHORT-NAME>
      <LONG-NAME>TID</LONG-NAME>
      <BYTE-POSITION>1</BYTE-POSITION>
      <TABLE-REF ID-REF="ES_OBDIICommonServices.
        TAB_ControlOfOnboardSystemsReadSupportedTIDs"/>
    </PARAM>
  </PARAMS>
</REQUEST>

<TABLE ID="ES_OBDIICommonServices.TAB_ControlOfOnboardSystemsReadSupportedTIDs">
  <SHORT-NAME>ControlOfOnboardSystemsReadSupportedTIDs</SHORT-NAME>
  <LONG-NAME>Control of Onboard Systems Read supported TIDs</LONG-NAME>
  <KEY-DOP-REF ID-REF="ES_OBDIICommonServices.DOP_TIDsSupported"/>
  <TABLE-ROW ID="TABROW_TID00TIDSupported01To20">
    <SHORT-NAME>TID00TIDSupported01To20</SHORT-NAME>
    <LONG-NAME>(TID 0x00) TID supported, 0x01 to 0x20</LONG-NAME>
    <KEY>0</KEY>
  </TABLE-ROW>
</TABLE>

```

```

        <STRUCTURE-REF ID-REF="STRUC_ControlOfOnboardSystemsTIDSupported01To20" />
    </TABLE-ROW>
</TABLE>

<STRUCTURE ID="STRUC_ControlOfOnboardSystemsTIDSupported01To20">
    <SHORT-NAME>ControlOfOnboardSystemsTIDSupported01To20</SHORT-NAME>
    <LONG-NAME>Control of Onboard Systems, TID supported, 0x01 to 0x20</LONG-NAME>
    <PARAMS>
        <PARAM SEMANTIC="DATA" xsi:type="VALUE">
            <SHORT-NAME>SupportedTIDs01To08</SHORT-NAME>
            <LONG-NAME>Supported TIDs0x01 to 0x08</LONG-NAME>
            <BYTE-POSITION>0</BYTE-POSITION>
            <DOP-REF ID-REF="_178" />
        </PARAM>
        ...
    </PARAMS>
</STRUCTURE>

<POS-RESPONSE ID="Resp_Service08RequestControlOfOnBoardSystemTestOrComponent">
    <SHORT-NAME>Service08RequestControlOfOnBoardSystemTestOrComponent</SHORT-NAME>
    <LONG-NAME>Service 0x08 - Request control of on-board system,
        test or component</LONG-NAME>
    <PARAMS>
        <PARAM xsi:type="CODED-CONST">
            <SHORT-NAME>RequestControlOfOnBoardSystem
                TestOrComponentResponseSID</SHORT-NAME>
            <LONG-NAME>Request control of on-board system,
                test or component response SID</LONG-NAME>
            <BYTE-POSITION>0</BYTE-POSITION>
            <CODED-VALUE>72</CODED-VALUE>
            <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
                <BIT-LENGTH>8</BIT-LENGTH>
            </DIAG-CODED-TYPE>
        </PARAM>
    </PARAMS>
</POS-RESPONSE>

```

7.6.8 Service 0x09 — Request vehicle information

7.6.8.1 ISO°15031-5 — Request vehicle information example

The request and response message described in this section can be used within ISO 15765-4.

Now the external test equipment requests the following InfoType:

— InfoType 0x02: VIN = [1G1JC5444R7252367] supported by ECU #1;

Table 37 — Request vehicle information request message

Message direction:		External test equipment → All ECUs	
Message Type:		Request	
Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Request vehicle information request SID	09	SIDRQ
#2	InfoType: 02 - VIN (Vehicle Identification Number)	02	INFTYP

Table 38 — Request vehicle information response message

Message direction:		ECU #1 → External test equipment	
Message Type:		Response	
Data Byte	Description (all values are in hexadecimal)	Byte Value (Hex)	Mnemonic
#1	Request vehicle information response SID	49	SIDPR
#2	InfoType: 02 - VIN (Vehicle Information Number)	02	INFTYP
#3	Number of data items: 01	01	NODI
#4	1 st ASCII character of VIN: '1'	31	VIN
#5	2 nd ASCII character of VIN: 'G'	47	VIN
#6	3 rd ASCII character of VIN: '1'	31	VIN
#7	4 th ASCII character of VIN: 'J'	4A	VIN
#8	5 th ASCII character of VIN: 'C'	43	VIN
#9	6 th ASCII character of VIN: '5'	35	VIN
#10	7 th ASCII character of VIN: '4'	34	VIN
#11	8 th ASCII character of VIN: '4'	34	VIN
#12	9 th ASCII character of VIN: '4'	34	VIN
#13	10 th ASCII character of VIN: 'R'	52	VIN
#14	11 th ASCII character of VIN: '7'	37	VIN
#15	12 th ASCII character of VIN: '2'	32	VIN
#16	13 th ASCII character of VIN: '5'	35	VIN
#17	14 th ASCII character of VIN: '2'	32	VIN
#18	15 th ASCII character of VIN: '3'	33	VIN
#19	16 th ASCII character of VIN: '6'	36	VIN
#20	17 th ASCII character of VIN: '7'	37	VIN

7.6.8.2 ISO 22901-2 — Request vehicle information in ODX sample

```

<DIAG-SERVICE ID="DiagnServi_Service09RequestVehicleInformation" ADDRESSING="FUNCTIONAL">
  <SHORT-NAME>Service09RequestVehicleInformation</SHORT-NAME>
  <LONG-NAME>Service 0x09 - Request vehicle information</LONG-NAME>
  <REQUEST-REF ID-REF="Req_Service09RequestVehicleInformation"/>
  <POS-RESPONSE-REFS>
    <POS-RESPONSE-REF ID-REF="Resp_Service09RequestVehicleInformation"/>
  </POS-RESPONSE-REFS>
</DIAG-SERVICE>

```

The following request is used to determine the supported Vehicle Information of an ECU

```

<REQUEST ID="Req_Service09RequestVehicleInformation">
  <SHORT-NAME>Service09RequestVehicleInformation</SHORT-NAME>
  <LONG-NAME>Service 0x09 - Request vehicle information</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="CODED-CONST">
      <SHORT-NAME>RequestVehicleInformationRequestSID</SHORT-NAME>
      <LONG-NAME>Request vehicle information request SID</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <CODED-VALUE>3</CODED-VALUE>
      <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
        <BIT-LENGTH>8</BIT-LENGTH>
      </DIAG-CODED-TYPE>
    </PARAM>
    <PARAM SEMANTIC="INFOTYPE" ID="Param_InfoType" xsi:type="TABLE-KEY">

```



```

        <SHORT-NAME>InfoType</SHORT-NAME>
        <LONG-NAME>InfoType</LONG-NAME>
        <BYTE-POSITION>1</BYTE-POSITION>
        <TABLE-REF ID-REF="ES_OBDIICommonServices.
            TAB_VehicleInformationReadSupportedInfoTypes"/>
    </PARAM>
</PARAMS>
</REQUEST>

<TABLE ID="ES_OBDIICommonServices.TAB_VehicleInformationReadSupportedInfoTypes">
    <SHORT-NAME>VehicleInformationReadSupportedInfoTypes</SHORT-NAME>
    <LONG-NAME>Vehicle Information Read supported InfoTypes</LONG-NAME>
    <KEY-DOP-REF ID-REF="_175"/>
    <TABLE-ROW ID="ES_OBDIICommonServices.
        TAB_VehicleInformationReadSupportedInfoTypes.
        TABROW_InfoType00InfoTypeSupported01To20">
        <SHORT-NAME>InfoType00InfoTypeSupported01To20</SHORT-NAME>
        <LONG-NAME>(InfoType 0x00) InfoType supported, 0x01 to 0x20</LONG-NAME>
        <KEY>0</KEY>
        <STRUCTURE-REF ID-REF="_219"/>
    </TABLE-ROW>
</TABLE>

```

The referenced STRUCTURE is then used to determine which information is supported.

```

<STRUCTURE ID="STRUC_SupportedInfoTypesInfoTypeSupported01To20">
    <SHORT-NAME>SupportedInfoTypesInfoTypeSupported01To20</SHORT-NAME>
    <LONG-NAME>Supported InfoTypes InfoType supported, 0x01 to 0x20</LONG-NAME>
    <PARAMS>
        <PARAM SEMANTIC="DATA" xsi:type="VALUE">
            <SHORT-NAME>_InfoType_00_InfoType_supported_01_20</SHORT-NAME>
            <LONG-NAME>Supported InfoTypes in 0x01 - 0x08</LONG-NAME>
            <BYTE-POSITION>0</BYTE-POSITION>
            <DOP-REF ID-REF="STRUC_SupportedInfoTypes01To08"/>
        </PARAM>
        ...
    </PARAMS>
</STRUCTURE>

<STRUCTURE ID="STRUC_SupportedInfoTypes01To08">
    <SHORT-NAME>SupportedInfoTypes01To08</SHORT-NAME>
    <LONG-NAME>Supported InfoTypes 0x01 to 0x08</LONG-NAME>
    <BYTE-SIZE>1</BYTE-SIZE>
    <PARAMS>
        <PARAM SEMANTIC="DATA" xsi:type="VALUE">
            <SHORT-NAME>InfoType0x08Supported</SHORT-NAME>
            <LONG-NAME>InfoType 0x08 supported</LONG-NAME>
            <BYTE-POSITION>0</BYTE-POSITION>
            <PHYSICAL-DEFAULT-VALUE>supported</PHYSICAL-DEFAULT-VALUE>
            <DOP-REF ID-REF="_16"/>
        </PARAM>
    </PARAMS>
</STRUCTURE>

```

When the supported Vehicle Information elements are determined, the following TABLE is used to determine the content.

```

<TABLE ID="ES_OBDIICommonServices.TAB_VehicleInformationReadInfoTypeData"
    SEMANTIC="INFOTYPE">
    <SHORT-NAME>VehicleInformationReadInfoTypeData</SHORT-NAME>
    <LONG-NAME>Vehicle Information Read InfoType Data</LONG-NAME>
    <KEY-DOP-REF ID-REF="ES_OBDIICommonServices.DOP_INFOTYPEsSupported"/>
    <TABLE-ROW ID="ES_OBDIICommonServices.
        TAB_Vehicle_nformationReadInfoTypeData.

```



```

        TABROW_InfoType02VehicleIdentificationNumber">
    <SHORT-NAME>InfoType02VehicleIdentificationNumber</SHORT-NAME>
    <LONG-NAME>(InfoType 0x02) Vehicle Identification Number</LONG-NAME>
    <KEY>2</KEY>
    <STRUCTURE-REF ID-REF="_224"/>
</TABLE-ROW>
<TABLE-ROW ID="ES_OBDIICommonServices.
        TAB_Vehicle_nformationReadInfoTypeData.
        TABROW_InfoType06CalibrationVerificationNumbers">
    <SHORT-NAME>InfoType06CalibrationVerificationNumbers</SHORT-NAME>
    <LONG-NAME>(InfoType 0x06) Calibration Verification Numbers</LONG-NAME>
    <KEY>6</KEY>
    <STRUCTURE-REF ID-REF="_173"/>
</TABLE-ROW>
</TABLE>

<POS-RESPONSE ID="Resp_Service09RequestVehicleInformation">
    <SHORT-NAME>Service09RequestVehicleInformation</SHORT-NAME>
    <LONG-NAME>Service 0x09 - Request vehicle information</LONG-NAME>
    <PARAMS>
        <PARAM xsi:type="CODED-CONST">
            <SHORT-NAME>RequestVehicleInformationResponseSID</SHORT-NAME>
            <LONG-NAME>Request vehicle information response SID</LONG-NAME>
            <BYTE-POSITION>0</BYTE-POSITION>
            <CODED-VALUE>73</CODED-VALUE>
            <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
                <BIT-LENGTH>8</BIT-LENGTH>
            </DIAG-CODED-TYPE>
        </PARAM>
        ...
    </PARAMS>
</POS-RESPONSE>

```

7.6.9 Authored data — PIDs

7.6.9.1 General

The PIDs are collected in the two tables `PIDsSupportedForPID11` and `PIDsSupportedForPID13`. In these tables a TABLE-ROW for each PID exists. The TABLE-ROW references the structured defined according to the rules in clause 7.4. The structure together with their DATA-OBJECT-PROPs for each PID is defined within ECU-SHARED-DATA `ObdIIDopsPIDs`. The following sub-clauses provide examples for each of the different value types.

7.6.9.2 ISO°15031-5 — Example "Linear"

Table 39 — PID 0x46 definition contains the definition of PID 0x46 as it can be found in ISO°15031-5. It falls into the value type category: Linear. The physical value in degrees Celsius can be expressed in the range — 40°C to +215°C. The encoded value uses the 8 bits of data byte A.

Table 39 — PID 0x46 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling/bit	External test equipment SI (Metric) / English display
46	Ambient air temperature (same scaling as IAT - \$0F)	A	– 40 °C	+ 215 °C	1 °C with – 40 °C offset	AAT: xxx °C / xxx °F
	AAT shall display ambient air temperature, if utilized by the control module strategy for OBD monitoring. AAT may be obtained directly from a sensor, may be obtained indirectly via the vehicle serial data communication bus, or may be inferred by the control strategy using other sensor inputs.					

7.6.9.3 ISO°22901-2 — Sample "Linear" value structure

For this PID the following TABLE-ROW might exists:

```
<TABLE-ROW ID="ES_OBDICommonServices.TAB_PIDsSupported.TABROW_OBDSupportedID46">
  <SHORT-NAME>OBDSupportedID46</SHORT-NAME>
  <LONG-NAME>OBD Supported ID 46</LONG-NAME>
  <KEY>70</KEY>
  <STRUCTURE-REF DOCTYPE="LAYER"
    ID-REF=" ES_ObdIIDopsPIDs.STR_PID46" />
</TABLE-ROW>
```

Please note that the key contains the value of the PID. As all numbers in ODX it is decimal.

The structure referenced from the TABLE-ROW reads as follows:

```
<STRUCTURE ID="ES_ObdIIDopsPIDs.STR_PID46">
  <SHORT-NAME>PID46</SHORT-NAME>
  <LONG-NAME>PID 46</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="VALUE">
      <SHORT-NAME>AmbientAirTemperature</SHORT-NAME>
      <LONG-NAME TI="OBD_PID46_AmbientAirTemperature">Ambient air temperature</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <DOP-SNREF SHORT-NAME="AmbientAirTemperatur" />
    </PARAM>
  </PARAMS>
</STRUCTURE>
```

It defines a parameter for the value and references the DOP for the encoding. That in turn can be generated along and reads as follows:

```
<DATA-OBJECT-PROP ID="ES_ObdIIDopsPIDs.DOP_AmbientAirTemperature">
  <SHORT-NAME>AmbientAirTemperature</SHORT-NAME>
  <LONG-NAME>Ambient air temperature</LONG-NAME>
  <COMPU-METHOD>
    <CATEGORY>LINEAR</CATEGORY>
    <COMPU-INTERNAL-TO-PHYS>
      <COMPU-SCALES>
        <COMPU-SCALE>
          <COMPU-RATIONAL-COEFFS>
```

```

        <COMPU-NUMERATOR><V>-40</V><V>1</V></COMPU-NUMERATOR>
        <COMPU-DENOMINATOR><V>1</V></COMPU-DENOMINATOR>
    </COMPU-RATIONAL-COEFFS>
    </COMPU-SCALE>
</COMPU-SCALES>
</COMPU-INTERNAL-TO-PHYS>
</COMPU-METHOD>
<DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
    <BIT-LENGTH>8</BIT-LENGTH>
</DIAG-CODED-TYPE>
<PHYSICAL-TYPE BASE-DATA-TYPE="A_INT32"/>
<UNIT-REF ID-REF="ES_ObdIIDopsPIDs.UNIT_DegreeCelsius"/>
</DATA-OBJECT-PROP>

```

It references some definition of the unit Celsius, but is otherwise complete. It formally defines the conversion to take the value of the 8 bits and subtract 40 from it to receive the value in degrees Celsius.

7.6.9.4 ISO°15031-5 — Example "Boolean"

The cutout of ISO°15031-5 shows an example of a value type with category Boolean. The “Malfunction indicator Lamp (MIL) Status” falls into this category. It can have only the values OFF or ON.

Table 40 — PID 0x01 definition

PID (hex)	Description	Data byte	Scaling/bit	External test equipment SI (Metric) / English display
01	Monitor status since DTCs cleared			
	The bits in this PID shall report two pieces of information for each monitor: — monitor status since DTCs were last cleared, saved in NVRAM or Keep Alive RAM; and — monitors supported on this vehicle.			
	Number of emission-related DTCs and MIL status	A (bit)	byte 1 of 4	DTC and MIL status:
	# of DTCs stored in this ECU	0-6	hex to decimal	DTC_CNT: xxd
	Malfunction Indicator Lamp (MIL) Status	7	0 = MIL OFF; 1 = MIL ON	MIL: OFF or ON
	The MIL status shall indicate “OFF” during the key-on, engine-off bulb check unless the MIL has also been commanded “ON” for a detected malfunction. The status should reflect whether there is any confirmed DTC(s) stored that are illuminating the MIL. It should not reflect the status of the MIL, which could be on for a function check, flashing I/M readiness or flashing for misfire.			

7.6.9.5 ISO°22901-2 — Sample "Boolean" value structure

The TABEL-ROW for PID 0x01 looks the same as the one for PID 0x46, with the following exceptions. It uses 1 as key and references another structure. The structure can be generated from the description above according to the Rules O1-O7. It results in:

```

<STRUCTURE ID="ES_ObdIIDopsPIDs.STR_PID01">
    <SHORT-NAME>PID01</SHORT-NAME>
    <LONG-NAME>PID 01</LONG-NAME>
    <PARAMS>
        <!--See Number examples for this parameter. -->
        <PARAM xsi:type="VALUE">
            <SHORT-NAME>MalfunctionIndicatorLampMILStatus</SHORT-NAME>
            <LONG-NAME TI="OBD_PID01_MalfunctionIndicatorLampMILStatus">Malfunction

```

```

Indicator Lamp (MIL) Status</LONG-NAME>
    <BYTE-POSITION>0</BYTE-POSITION>
    <BIT-POSITION>7</BIT-POSITION>
    <DOP-SNREF SHORT-NAME="1BitOnOff"/>
</PARAM>
    <!-- Additional parameters of PID 01 go here. -->
</PARAMS>
</STRUCTURE>

```

The structure contains the reference to the location in the PDU and defines the name according to rule O2. The conversion is part of the referenced DATA-OBJECT-PROP:

```

<DATA-OBJECT-PROP ID="ES_ObdIIDopsPIDs.DOP_1BitOnOff">
    <SHORT-NAME>1BitOnOff</SHORT-NAME>
    <LONG-NAME>1Bit On Off</LONG-NAME>
    <COMPU-METHOD>
        <CATEGORY>TEXTTABLE</CATEGORY>
        <COMPU-INTERNAL-TO-PHYS>
            <COMPU-SCALES>
                <COMPU-SCALE>
                    <LOWER-LIMIT>0</LOWER-LIMIT>
                    <COMPU-CONST><VT>OFF</VT></COMPU-CONST>
                </COMPU-SCALE>
                <COMPU-SCALE>
                    <LOWER-LIMIT>1</LOWER-LIMIT>
                    <COMPU-CONST><VT>ON</VT></COMPU-CONST>
                </COMPU-SCALE>
            </COMPU-SCALES>
        </COMPU-INTERNAL-TO-PHYS>
    </COMPU-METHOD>
    <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32"
        xsi:type="STANDARD-LENGTH-TYPE">
        <BIT-LENGTH>1</BIT-LENGTH>
    </DIAG-CODED-TYPE>
    <PHYSICAL-TYPE BASE-DATA-TYPE="A_UNICODE2STRING"/>
</DATA-OBJECT-PROP>

```

The single bit that constitutes this value is interpreted as a number resulting in the value 0 and 1. This is used as index in a TEXTTABLE that contains the physical textual values “OFF” and “ON”. Please note that this DATA-OBJECT-PROP can and should be reused for all values, that use this encoding and texts.

7.6.9.6 ISO¹⁵⁰³¹⁻⁵ — Example "BitSelect"

Table 41 — PID 0x03 definition shows "Fuel system 1 status" an example of the value type "BitSelect". It uses data byte A.

Table 41 — PID 0x03 definition

PID (hex)	Description	Data Byte	Scaling/bit	External test equipment SI (Metric)/English display
03	Fuel system 1 status:	A (bit)	byte 1 of 2	FUELSYS1:
	(Unused bits shall be reported as '0'; no more than one bit at a time can be set to a '1' of that bank.)	0	1 = Open loop - has not yet satisfied conditions to go closed loop	OL
		1	1 = Closed loop - using oxygen sensor(s) as feedback for fuel control	CL
		2	1 = Open loop due to driving conditions (e.g. power enrichment, deceleration enrichment)	OL-Drive
		3	1 = Open loop - due to detected system fault	OL-Fault
		4	1 = Closed loop, but fault with at least one oxygen sensor - may be using single oxygen sensor for fuel control	CL-Fault
		5-7	ISO/SAE reserved (bits shall be reported as '0')	—
NOTE Fuel systems do not normally refer to injector banks. Fuel systems are intended to represent completely different fuel systems that can independently enter and exit closed-loop fuel. Banks of injectors on a V-engine are generally not independent and share the same closed-loop enablement criteria. If the engine is off and the ignition is on, all bits in Data Byte A and Data Byte B shall be reported as '0'.				

7.6.9.7 ISO²²⁹⁰¹⁻² — Sample "BitSelect" value structure

Again the TABLE-ROW differs only in a standard way from the example already given above. The referenced structure, created according to the rules B1-B6, looks the following:

```
<STRUCTURE ID="ES_ObdIIDopsPIDs.STR_PID03">
  <SHORT-NAME>PID03</SHORT-NAME>
  <LONG-NAME>PID 03</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="VALUE">
      <SHORT-NAME>FuelSystem1Status</SHORT-NAME>
      <LONG-NAME TI="OBD_PID03_FuelSystem1Status">Fuel system 1 status:</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <DOP-SNREF SHORT-NAME="FuelSystemBitSelect"/>
    </PARAM>
    <PARAM xsi:type="VALUE">
      <SHORT-NAME>FuelSystem2Status</SHORT-NAME>
      <LONG-NAME TI="OBD_PID03_FuelSystem2Status">Fuel system 2 status:</LONG-NAME>
      <BYTE-POSITION>1</BYTE-POSITION>
      <DOP-SNREF SHORT-NAME="FuelSystemBitSelect"/>
    </PARAM>
  </PARAMS>
</STRUCTURE>
```

A parameter, that defines names and location in the PDU, is defined for each value of that PID. The encoding is defined in the DATA-OBJECT-PROP. As both values of the PID 03 use the same encoding the DATA-OBJECT-PROP is reused.

It is defined as follows:

```
<DATA-OBJECT-PROP ID="ES_ObdIIDopsPIDs.DOP_FuelSystemBitSelect">
  <SHORT-NAME>FuelSystemBitSelect</SHORT-NAME>
  <LONG-NAME>Fuel System BitSelect</LONG-NAME>
  <COMPU-METHOD>
    <CATEGORY>TEXTTABLE</CATEGORY>
    <COMPU-INTERNAL-TO-PHYS>
      <COMPU-SCALES>
        <COMPU-SCALE>
          <LOWER-LIMIT>0</LOWER-LIMIT>
          <COMPU-CONST><VT>-</VT></COMPU-CONST>
        </COMPU-SCALE>
        <COMPU-SCALE>
          <LOWER-LIMIT>1</LOWER-LIMIT>
          <COMPU-CONST><VT>OL</VT></COMPU-CONST>
        </COMPU-SCALE>
        <COMPU-SCALE>
          <LOWER-LIMIT>2</LOWER-LIMIT>
          <COMPU-CONST><VT>CL</VT></COMPU-CONST>
        </COMPU-SCALE>
        <COMPU-SCALE>
          <LOWER-LIMIT>4</LOWER-LIMIT>
          <COMPU-CONST><VT>OL-Drive</VT></COMPU-CONST>
        </COMPU-SCALE>
        <COMPU-SCALE>
          <LOWER-LIMIT>8</LOWER-LIMIT>
          <COMPU-CONST><VT>OL-Fault</VT></COMPU-CONST>
        </COMPU-SCALE>
        <COMPU-SCALE>
          <LOWER-LIMIT>16</LOWER-LIMIT>
          <COMPU-CONST><VT>CL-Fault</VT></COMPU-CONST>
        </COMPU-SCALE>
      </COMPU-SCALES>
    </COMPU-INTERNAL-TO-PHYS>
  </COMPU-METHOD>
  <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UNIT32"
    xsi:type="STANDARD-LENGTH-TYPE">
    <BIT-LENGTH>8</BIT-LENGTH></DIAG-CODED-TYPE>
  <PHYSICAL-TYPE BASE-DATA-TYPE="A_UNICODE2STRING" />
</DATA-OBJECT-PROP>
```

The bitfield used for the value is interpreted as a number that is used in the TEXTTABLE as key. As only one bit can be set at any time, only the powers of two have entries. This encoding ensures, that the D-Server will signal an error if more than one bit is set.

7.6.9.8 ISO°15031-5 — Example "BitSet"

Table 42 — PID 0x13 definition shows “Location of Oxygen Sensors” an example of the value type “Bitset”. It uses data byte A. Each bit that is set, represents the existence of one of the oxygen sensors.

Table 42 — PID 0x13 definition

PID (hex)	Description	Data byte	Scaling/bit	External test equipment SI (Metric) / English display
13	Location of Oxygen Sensors	A (bit)	byte 1 of 1	O2SLOC:
	(Where sensor 1 is closest to the engine. Each bit indicates the presence or absence of an oxygen sensor at the following location.)	0	1 = Bank 1 - Sensor 1 present at that location	O2S11
		1	1 = Bank 1 - Sensor 2 present at that location	O2S12
		2	1 = Bank 1 - Sensor 3 present at that location	O2S13
		3	1 = Bank 1 - Sensor 4 present at that location	O2S14
		4	1 = Bank 2 - Sensor 1 present at that location	O2S21
		5	1 = Bank 2 - Sensor 2 present at that location	O2S22
		6	1 = Bank 2 - Sensor 3 present at that location	O2S23
		7	1 = Bank 2 - Sensor 4 present at that location	O2S24

7.6.9.9 ISO°22901-2 — Sample "BitSet" value structure

Again the TABLE-ROW differs only in a standard way from the example already given above. The referenced structure, created according to the rules S1-S7, looks the following:

```
<STRUCTURE ID="ES_ObdIIDopsPIDs.STR_PID13">
  <SHORT-NAME>PID13</SHORT-NAME>
  <LONG-NAME>PID 13</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="VALUE">
      <SHORT-NAME>LocationOfOxygenSensorsBank1Sensor1</SHORT-NAME>
      <LONG-NAME TI="OBD_PID13_LocationOfOxygenSensorsBank1Sensor1">Location of
Oxygen Sensors Bank 1 - Sensor 1</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <BIT-POSITION>0</BIT-POSITION>
      <DOP-SNREF SHORT-NAME="LocationOfOxygenSensorsO2S11"/>
    </PARAM>
    <PARAM xsi:type="VALUE">
      <SHORT-NAME>LocationOfOxygenSensorsBank1Sensor2</SHORT-NAME>
      <LONG-NAME TI="OBD_PID13_LocationOfOxygenSensorsBank1Sensor2">Location of
Oxygen Sensors Bank 1 - Sensor 2</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <BIT-POSITION>1</BIT-POSITION>
      <DOP-SNREF SHORT-NAME="LocationOfOxygenSensorsO2S12"/>
    </PARAM>
    <PARAM xsi:type="VALUE">
      <SHORT-NAME>LocationOfOxygenSensorsBank1Sensor3</SHORT-NAME>
      <LONG-NAME TI="OBD_PID13_LocationOfOxygenSensorsBank1Sensor3">Location of
Oxygen Sensors Bank 1 - Sensor 3</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <BIT-POSITION>2</BIT-POSITION>
```

```

        <DOP-SNREF SHORT-NAME="LocationOfOxygenSensorsO2S13" />
    </PARAM>
    <!-- Additional parameters go here. -->
</PARAMS>

</STRUCTURE>

```

The names are derived according to rule S2. The part “present at location” of the “Scaling/bit” columns is considered not deterministic. While “Location of Oxygen Sensors” is the real name of these groups of parameters each one must be distinguishable and thus contains an addition describing bank and sensor number. As the value of the parameter is different for each sensor, according to rule S7, for each parameter a different DATA-OBJECT-PROP is defined.

As they all follow, of course, the same structure only the first is given here:

```

<DATA-OBJECT-PROP ID="ES_ObdIIDopsPIDs.DOP_LocationOfOxygenSensorsO2S11">
    <SHORT-NAME>LocationOfOxygenSensorsO2S11</SHORT-NAME>
    <LONG-NAME>Location of Oxygen Sensors O2S11</LONG-NAME>
    <COMPU-METHOD>
        <CATEGORY>TEXTTABLE</CATEGORY>
        <COMPU-INTERNAL-TO-PHYS>
            <COMPU-SCALES>
                <COMPU-SCALE>
                    <LOWER-LIMIT>0</LOWER-LIMIT>
                    <COMPU-CONST><VT></VT></COMPU-CONST>
                </COMPU-SCALE>
                <COMPU-SCALE>
                    <LOWER-LIMIT>1</LOWER-LIMIT>
                    <COMPU-CONST><VT>O2S11</VT></COMPU-CONST>
                </COMPU-SCALE>
            </COMPU-SCALES>
        </COMPU-INTERNAL-TO-PHYS>
    </COMPU-METHOD>
    <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32"
        xsi:type="STANDARD-LENGTH-TYPE">
        <BIT-LENGTH>1</BIT-LENGTH>
    </DIAG-CODED-TYPE>
    <PHYSICAL-TYPE BASE-DATA-TYPE="A_UNICODE2STRING" />
</DATA-OBJECT-PROP>

```

The conversion structure resembles that of the “Boolean” value type. Yet, here 0 is mapped to the empty text, while 1 is mapped to the specific value defined in the “English display” column.

7.6.9.10 ISO 15031-5 — Example "Number2Text"

Table 43 — PID 0x1C definition shows “OBD requirements to which vehicle is designed” as an example of the value type “Number2Text”. It matches different numbers 0x1, 0x2, ...0xFF to various texts shown in the “SI (Metric) / English display” column. For example, the number 10 (0xA) is mapped to the text “JOB D”.

Table 43 — PID 0x1C definition

PID (hex)	Description	Data byte	Scaling	External test equipment SI (Metric) / English display
1C	OBD requirements to which vehicle is designed	A (hex)	byte 1 of 1 (State Encoded Variable)	OBDSUP:
	OBD II (California ARB)	01		OBD II
	OBD (Federal EPA)	02		OBD
	OBD and OBD II	03		OBD and OBD II
	OBD I	04		OBD I
	Not OBD compliant	05		NO OBD
	EOBD	06		EOBD
	EOBD and OBD II	07		EOBD and OBD II
	EOBD and OBD	08		EOBD and OBD
	EOBD, OBD and OBD II	09		EOBD, OBD and OBD II
	JOB D	0A		JOB D
	JOB D and OBD II	0B		JOB D and OBD II
	JOB D and EOBD	0C		JOB D and EOBD
	JOB D, EOBD, and OBD II	0D		JOB D, EOBD, and OBD II
	Heavy Duty Vehicles (EURO IV) B1	0E		EURO IV B1
	Heavy Duty Vehicles (EURO V) B2	0F		EURO V B2
	Heavy Duty Vehicles (EURO EEC) C (gas engines)	10		EURO C
	Engine Manufacturer Diagnostics (EMD)	11		EMD
	ISO/SAE reserved	12 - FA		—
	ISO/SAE - Not available for assignment	FB - FF		SAE J1939 special meaning

7.6.9.11 ISO 22901-2 — Sample "Number2Text" value structure

The TABLE-ROW differs only in a standard way from the example already given above. The referenced structure, created according to the rules X1-X6, is:

```
<STRUCTURE ID="ES_ObdIIDopsPIDs.STR_PID1C">
  <SHORT-NAME>PID1C</SHORT-NAME>
  <LONG-NAME>PID 1C</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="VALUE">
      <SHORT-NAME>OBDRequirementsToWhichVehicleIsDesigned</SHORT-NAME>
      <LONG-NAME TI="OBD_PID1C_OBDRequirementsToWhichVehicleIsDesigned">OBD
requirements to which vehicle is designed</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <DOP-SNREF SHORT-NAME="OBDRequirements" />
    </PARAM>
  </PARAMS>
</STRUCTURE>
```

```

</PARAMS>
</STRUCTURE>

```

The parameter in the structure contains not much information. The whole mapping takes place in the DATA-OBJECT-PROP:

```

<DATA-OBJECT-PROP ID="ES_ObdIIDopsPIDs.DOP_OBDRequirements">
  <SHORT-NAME>OBDRequirements</SHORT-NAME>
  <LONG-NAME>OBD Requirements</LONG-NAME>
  <COMPU-METHOD>
    <CATEGORY>TEXTTABLE</CATEGORY>
    <COMPU-INTERNAL-TO-PHYS>
      <COMPU-SCALES>
        <COMPU-SCALE>
          <LOWER-LIMIT>1</LOWER-LIMIT>
          <COMPU-CONST><VT>OBD II</VT></COMPU-CONST>
        </COMPU-SCALE>
        <COMPU-SCALE>
          <LOWER-LIMIT>2</LOWER-LIMIT>
          <COMPU-CONST><VT>OBD</VT></COMPU-CONST>
        </COMPU-SCALE>
        <COMPU-SCALE>
          <LOWER-LIMIT>3</LOWER-LIMIT>
          <COMPU-CONST><VT>OBD and OBD II</VT></COMPU-CONST>
        </COMPU-SCALE>
        <COMPU-SCALE>
          <LOWER-LIMIT>4</LOWER-LIMIT>
          <COMPU-CONST><VT>OBD I</VT></COMPU-CONST>
        </COMPU-SCALE>
        <COMPU-SCALE>
          <LOWER-LIMIT>5</LOWER-LIMIT>
          <COMPU-CONST><VT>NO OBD</VT></COMPU-CONST>
        </COMPU-SCALE>
        <COMPU-SCALE>
          <LOWER-LIMIT>6</LOWER-LIMIT>
          <COMPU-CONST><VT>EOBD</VT></COMPU-CONST>
        </COMPU-SCALE>
        <!-- ... -->
        <COMPU-SCALE>
          <LOWER-LIMIT>10</LOWER-LIMIT>
          <COMPU-CONST><VT>JOB</VT></COMPU-CONST>
        </COMPU-SCALE>
        <!-- ... -->
        <COMPU-SCALE>
          <LOWER-LIMIT>18</LOWER-LIMIT>
          <UPPER-LIMIT>250</UPPER-LIMIT>
          <COMPU-CONST><VT>-</VT></COMPU-CONST>
        </COMPU-SCALE>
        <COMPU-SCALE>
          <LOWER-LIMIT>251</LOWER-LIMIT>
          <UPPER-LIMIT>255</UPPER-LIMIT>
          <COMPU-CONST><VT>SAE J1939 special meaning</VT></COMPU-CONST>
        </COMPU-SCALE>
      </COMPU-SCALES>
    </COMPU-INTERNAL-TO-PHYS>
  </COMPU-METHOD>
  <DIAG-CODED-TYPE xsi:type="STANDARD-LENGTH-TYPE" BASE-DATA-TYPE="A_UINT32">
    <BIT-LENGTH>8</BIT-LENGTH>
  </DIAG-CODED-TYPE>

```

```
<PHYSICAL-TYPE BASE-DATA-TYPE="A_UNICODE2STRING"></PHYSICAL-TYPE>
</DATA-OBJECT-PROP>
```

The DIAG-CODED-TYPE defines the number interpretation of the 8bits. The COMPU-METHOD, as TEXTTABLE, maps these numbers to texts. The TEXTTABLE shown here does not contain all entries. The others are similar.

7.6.9.12 ISO¹⁵⁰³¹⁻⁵ — Example "Number"

ISO¹⁵⁰³¹⁻⁵ defines the number of DTCs as part of the PID 01 definition. This value is of type Number.

7.6.9.13 ISO²²⁹⁰¹⁻² — Sample "Number" value structure

Again the TABLE-ROW differs only in a standard way from the example already given above. The referenced structure, created according to the rules N1-N7, looks the following:

```
<STRUCTURE ID="ES_ObdIIDopsPIDs.STR_PID01">
  <SHORT-NAME>PID01</SHORT-NAME>
  <LONG-NAME>PID 01</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="VALUE">
      <SHORT-NAME>OfDTCsStoredInThisECU</SHORT-NAME>
      <LONG-NAME TI="OBD_PID01_OfDTCsStoredInThisECU"># of DTCs stored in this
ECU</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <DOP-SNREF SHORT-NAME="7BitIdentical"/>
    </PARAM>
    <!-- Additional parameters of PID 01 go here. -->
  </PARAMS>
</STRUCTURE>
```

The example code now shows the first value that is returned in byte A of the PID.

```
<DATA-OBJECT-PROP ID="ES_ObdIIDopsPIDs.DOP_7BitIdentical">
  <SHORT-NAME>7BitIdentical</SHORT-NAME>
  <LONG-NAME>7 Bit Identical</LONG-NAME>
  <COMPU-METHOD>
    <CATEGORY>IDENTICAL</CATEGORY>
  </COMPU-METHOD>
  <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
    <BIT-LENGTH>7</BIT-LENGTH>
  </DIAG-CODED-TYPE>
  <PHYSICAL-TYPE BASE-DATA-TYPE="A_UINT32"/>
</DATA-OBJECT-PROP>
```

The DATA-OBJECT-PROP defines to interpret the 7 bits as an unsigned number.

7.6.9.14 ISO°15031-5 — Example "DTC"

Table 44 — PID 0x02 definition defines PID 0x02. Its only value is a reference to a DTC. These are, according to clause 7.5 defined in ECU-SHARED-DATA ObdIIDopsDtcDeclarations.

Table 44 — PID 0x02 definition

PID (hex)	Description	Data byte	Min. value	Max. value	Scaling	External test equipment SI (Metric) / English display
02	DTC that caused required freeze frame data storage (\$0000 indicates no freeze frame data)	A, B	00 00	FF FF	Hexadecimal e.g. P01AB (DTCs defined in ISO 15031-§6)	DTCFRZF: Pxxxx, Cxxxx, Bxxxx, Uxxxx

7.6.9.15 ISO°22901-2 — Sample "DTC" value structure

Neither TABLE-ROW nor STRUCTURE for PID 02 are very complicated:

```
<STRUCTURE ID="ES_ObdIIDopsPIDs.STR_PID02">
  <SHORT-NAME>PID02</SHORT-NAME>
  <LONG-NAME>PID 02</LONG-NAME>
  <PARAMS>
    <PARAM xsi:type="VALUE">
      <SHORT-NAME>DTCThatCausedRequiredFreezeFrameDataStorage</SHORT-NAME>
      <LONG-NAME TI="OBD_PID02_DTCThatCausedRequiredFreezeFrameDataStorage">DTC that
caused required freeze frame data storage</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <DOP-SNREF SHORT-NAME="ObdDtcs" />
    </PARAM>
  </PARAMS>
</STRUCTURE>
```

While the referenced DOP is defined in an ECU-SHARED-DATA other than ObdIIDopsPIDs, it can still be referenced by SHORT-NAME, as ECU-SHARED-DATA "ObdIIDopsDtcDeclarations". The DATA-OBJECT-PROP itself will be shown in the following clause 7.6.10.

7.6.10 Authored data — DTCs

DTCs in ODX are authored in DTC-DOPs. They define how the code itself is extracted from the PDU. In addition they list all diagnostic trouble codes and their description. According to the rules in clause 7.5 the following DTC-DOP for powertrain DTCs will result (cut-out):

```
<DTC-DOP ID="ES_ObdIIDopsDtcDeclarations.DOP_ObdDtcsPowertrain">
  <SHORT-NAME>ObdDtcsPowertrain</SHORT-NAME>
  <DIAG-CODED-TYPE BASE-DATA-TYPE="A_UINT32" xsi:type="STANDARD-LENGTH-TYPE">
    <BIT-LENGTH>16</BIT-LENGTH>
  </DIAG-CODED-TYPE>
  <PHYSICAL-TYPE BASE-DATA-TYPE="A_UINT32" DISPLAY-RADIX="HEX" />
  <COMPU-METHOD>
    <CATEGORY>IDENTICAL</CATEGORY>
  </COMPU-METHOD>
  <DTCS>
    <DTC ID="ObdDtcs_P0000">
      <SHORT-NAME>ObdDtcs_P0000</SHORT-NAME>
      <TROUBLE-CODE>0</TROUBLE-CODE>
```

```

        <DISPLAY-TROUBLE-CODE>P0000</DISPLAY-TROUBLE-CODE>
        <TEXT TI="ObdDtc_P0000">ISO/SAE reserved - Use Not Allowed</TEXT>
    </DTC>
    <DTC ID="ObdDtc_P0001">
        <SHORT-NAME>ObdDtc_P0001</SHORT-NAME>
        <TROUBLE-CODE>1</TROUBLE-CODE>
        <DISPLAY-TROUBLE-CODE>P0001</DISPLAY-TROUBLE-CODE>
        <TEXT TI="ObdDtc_P0001">Fuel Volume Regulator Control Circuit/Open</TEXT>
    </DTC>
    <!-- Other DTCs go here -->
    <DTC ID="ObdDtc_P0008">
        <SHORT-NAME>ObdDtc_P0008</SHORT-NAME>
        <TROUBLE-CODE>8</TROUBLE-CODE>
        <DISPLAY-TROUBLE-CODE>P0008</DISPLAY-TROUBLE-CODE>
        <TEXT TI="ObdDtc_P0008">Engine Position System Performance Bank 1</TEXT>
    </DTC>
    <DTC ID="ObdDtc_P0009">
        <SHORT-NAME>ObdDtc_P0009</SHORT-NAME>
        <TROUBLE-CODE>9</TROUBLE-CODE>
        <DISPLAY-TROUBLE-CODE>P0009</DISPLAY-TROUBLE-CODE>
        <TEXT TI="ObdDtc_P0009">Engine Position System Performance Bank 2</TEXT>
    </DTC>
    <DTC ID="ObdDtc_P000A">
        <SHORT-NAME>ObdDtc_P000A</SHORT-NAME>
        <TROUBLE-CODE>10</TROUBLE-CODE>
        <DISPLAY-TROUBLE-CODE>P000A</DISPLAY-TROUBLE-CODE>
        <TEXT TI="ObdDtc_P000A">"A" Camshaft Position Slow Response Bank 1</TEXT>
        <!-- a) The "A" camshaft shall be either the "intake,"
             "left," or "front" camshaft. Left/Right and Front/Rear
             are determined as if viewed from the driver's seating position.
             Bank 1 contains cylinder number one, Bank 2 is the opposite bank. -->
    </DTC>
    <!-- Further Powertrain DTCs go here -->
</DTCS>
</DTC-DOP>

```

The conversion of the trouble code is straight forward. It is an unsigned 16bit number in binary encoding. The resulting value is matched against the entry in TROUBLE-CODE. It is given in decimal, while all references to it in names and display use SAE J2012 notation. Please note also, that comments have been converted to XML comments.

For the other groups (Chassis, Body, and Network) similar DTC-DOPs are defined. They are united in a single DTC-DOP with SHORT-NAME ObdDtc:

```

<DTC-DOP ID="ObdDtc cs">
    <SHORT-NAME>ObdDtc cs</SHORT-NAME>
    <DIAG-CODED-TYPE xsi:type="STANDARD-LENGTH-TYPE" BASE-DATA-TYPE="A_UINT32">
        <BIT-LENGTH>16</BIT-LENGTH></DIAG-CODED-TYPE>
    <PHYSICAL-TYPE BASE-DATA-TYPE="A_UINT32" DISPLAY-RADIX="HEX" />
    <COMPU-METHOD>
        <CATEGORY>IDENTICAL</CATEGORY>
    </COMPU-METHOD>
    <DTCS>
        <DTC-REF ID-REF="ObdDtc_P0000" />
    </DTCS>
    <LINKED-DTC-DOPS>
        <LINKED-DTC-DOP>
            <NOT-INHERITED-DTC-SNREFS>
                <NOT-INHERITED-DTC-SNREF SHORT-NAME="ObdDtc_P0000" />
            </NOT-INHERITED-DTC-SNREFS>
        </LINKED-DTC-DOP>
    </LINKED-DTC-DOPS>
</DTC-DOP>

```

```
<DTC-DOP-REF ID-REF="ObdDtcsPowertrain" />
</LINKED-DTC-DOP>
<LINKED-DTC-DOP>
  <DTC-DOP-REF ID-REF="ObdDtcsNetwork" />
</LINKED-DTC-DOP>
<LINKED-DTC-DOP>
  <DTC-DOP-REF ID-REF="ObdDtcsChassis" />
</LINKED-DTC-DOP>

</LINKED-DTC-DOPS>

</DTC-DOP>
```

This DTC-DOP is referenced from PID values that use the DTC value type.