INTERNATIONAL STANDARD

ISO 15031-5

Third edition
2015-08-01

Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics —

Part 5: Emissions-related diagnostic services

Véhicules routiers — Communications entre un véhicule et un équipement externe pour le diagnostic relatif aux émissions —

Partie 5: Services de diagnostic relatif aux émissions

Reference number
ISO 15031-5:2015(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/TC 22, *Road vehicles*, Subcommittee SC 31, *Data communication.*

This third edition cancels and replaces the second edition (ISO 15031-5:2011), which has been technically revised.

ISO 15031 consists of the following parts, under the general title *Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics*:

— *Part 1: General information and use case definition*

— *Part 2: Guidance on terms, definitions, abbreviations and acronyms*

— *Part 3: Diagnostic connector and related electrical circuits, specification and use*

— *Part 4: External test equipment*

— *Part 5: Emissions-related diagnostic services*

— *Part 6: Diagnostic trouble code definitions*

— *Part 7: Data link security*

Introduction

Overview

ISO 15031 consists of a number of parts which, taken together, provide a coherent self-consistent set of specifications to facilitate emissions-related diagnostics. ISO 15031-1 provides an introduction to the series of International Standards. ISO 15031-2 through ISO 15031-7 are based on SAE recommended practices. This part of ISO 15031 is based on SAE J1979.

This International Standard includes the communication between the vehicle's On-Board Diagnostic (OBD) systems and test equipment implemented across vehicles within the scope of the legislated emissions-related OBD.

To achieve this, it is based on the Open Systems Interconnection (OSI) Basic Reference Model in accordance with ISO/IEC 7498-1 and ISO/IEC 10731, which structures communication systems into seven layers. When mapped on this model, the services specified by this International Standard are broken into the following layers in accordance with Table 1.

— Diagnostic services (layer 7), specified in the following:
  — this part of ISO 15031;
  — ISO 27145-3 (WWH-OBD).
— Presentation layer (layer 6), specified in the following:
  — ISO 15031-2, SAE J1930-DA;
  — this part of ISO 15031, SAE J1979-DA;
  — ISO 15031-6, SAE J2012-DA;
  — ISO 27145-2, SAE J2012-DA.
— Session layer services (layer 5), specified in the following:
  — ISO 14229-2 supports ISO 15765-4 DoCAN and ISO 14230-4 DoK-Line protocols;
  — ISO 14229-2 is not applicable to the SAE J1850 and ISO 9141-2 protocols.
— Transport layer services (layer 4), specified in the following:
  — ISO 15765-2;
  — SAE J1850 defined in this part of ISO 15031;
  — ISO 9141-2 defined in this part of ISO 15031;
  — ISO 14230-4, defined in this part of ISO 15031.
— Network layer services (layer 3), specified in the following:
  — ISO 15765-2;
  — SAE J1850 defined in this part of ISO 15031;
  — ISO 9141-2 defined in this part of ISO 15031;
  — ISO 14230-4 defined in this part of ISO 15031.
— Data link layer (layer 2), specified in the following:
  — ISO 15765-4, ISO 11898-1, and ISO 11898-2;

- SAE J1850;
- ISO 9141-2;
- ISO 14230-2.

— Physical layer (layer 1), specified in the following:

- ISO 15765-4, ISO 11898-1, and ISO 11898-2;
- SAE J1850;
- ISO 9141-2;
- ISO 14230-1.

Table 1 — Legislated emissions-related OBD/WWH[a]-OBD diagnostic specifications applicable to the OSI layers

<table>
<tr><th>Applicability</th><th>OSI 7 layers</th><th colspan="6">Emissions-related OBD communication requirements</th><th colspan="3">Emissions-related WWH-OBD communication requirements</th></tr>
<tr><td rowspan="8">Seven layers according to ISO/IEC 7498-1 and ISO/IEC 10731</td><td>Application (layer 7)</td><td colspan="6">ISO 15031-5/SAE J1979</td><td colspan="3">ISO 27145-3</td></tr>
<tr><td rowspan="2">Presentation (layer 6)</td><td colspan="6">ISO 15031-2, ISO 15031-5, ISO 15031-6</td><td colspan="3">ISO 27145-2</td></tr>
<tr><td colspan="6">SAE J1930-DA, SAE J1979-DA, SAE J2012-DA</td><td colspan="3">SAE J1930-DA, SAE J1979-DA, SAE J2012-DA</td></tr>
<tr><td>Session (layer 5)</td><td colspan="2">Not applicable</td><td colspan="4">ISO 14229-2</td></tr>
<tr><td>Transport (layer 4)</td><td colspan="2" rowspan="2">ISO 15031-5</td><td rowspan="2">ISO 14230-4</td><td rowspan="2">ISO 15765-2</td><td rowspan="3">ISO 15765-4</td><td rowspan="2">ISO 15765-2</td><td rowspan="3">ISO 15765-4</td><td rowspan="2">ISO 13400-2</td></tr>
<tr><td>Network (layer 3)</td></tr>
<tr><td>Data link (layer 2)</td><td rowspan="2">SAE J1850</td><td rowspan="2">ISO 9141-2</td><td>ISO 14230-2</td><td rowspan="2">ISO 11898-1, ISO 11898-2</td><td rowspan="2">ISO 11898-1, ISO 11898-2</td><td rowspan="2">ISO 13400-3</td></tr>
<tr><td>Physical (layer 1)</td><td>ISO 14230-1</td></tr>
<tr><td colspan="10">[a] World-Wide Harmonized.</td></tr>
</table>

SAE document reference concept

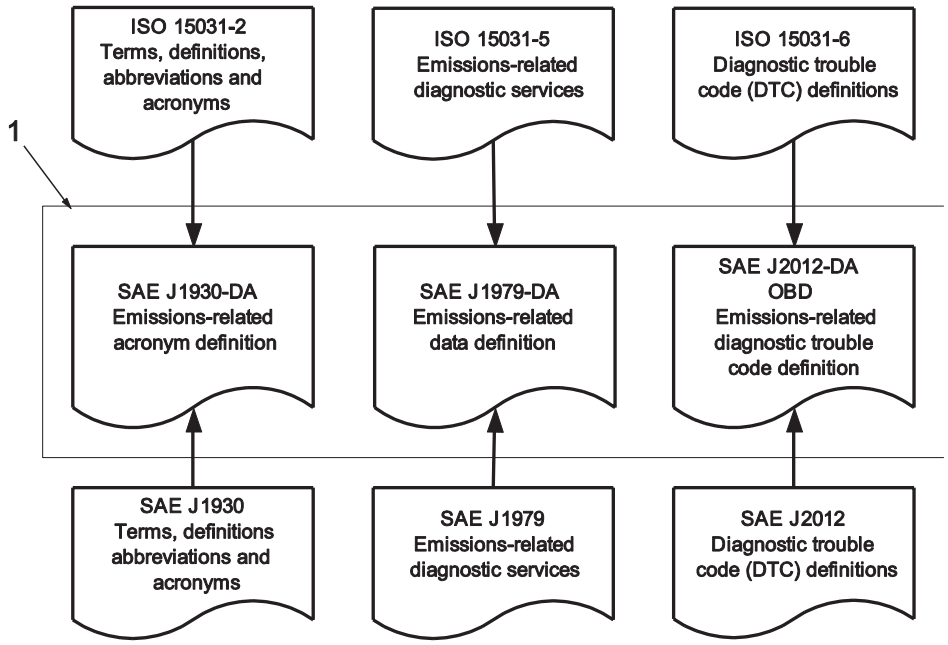
ISO 15031 references several SAE documents which contain all terms, data, and DTC (diagnostic trouble code) definitions. This is illustrated in Figure 1.

Additional information on the content of the referenced documents is given below:

— SAE J1930: the document is concerned with a procedure for naming objects and systems and with the set of words from which names are built. It references SAE J1930-DA which contains all standardized naming objects, terms, and abbreviations.

— SAE J1979: the document is concerned with the definition of emissions-related diagnostic services (diagnostic test modes). It references SAE J1979-DA which contains all standardized data items such as PIDs, Test IDs, Monitor IDs, and INFOTYPE IDs.

— SAE J2012: the document is concerned with the procedure for defining emissions-related DTCs. It references SAE J2012-DA which contains all standardized data items such as DTCs and FTBs (failure type bytes).



Key

1 SAE Digital Annexes

Figure 1 — SAE Digital Annex document reference

OBD regulations require passenger cars and light, medium, and heavy duty trucks to support a minimum set of diagnostic information to external (off-board) "generic" test equipment.

SAE J1979-DA (OBD) Digital Annex

This part of ISO 15031 references SAE J1979-DA. SAE J1979-DA is concerned with the definition of the following:

— Parameter Identifiers (PIDs);

— Test IDentifiers (TIDs);

— OBD Monitor Identifiers (OBDMIDs);

— Unit and Scaling Identifiers (UASIDs);

— INFOTYPEs (INFOTYPEs).

SAE Digital Annex revision procedure

New emissions-related regulatory requirements drive new in-vehicle technology to lower emissions. New technology-related OBD monitor data and DTCs need to be standardized to support the external (off-board) "generic" test equipment. All relevant information is proposed by the automotive industry represented by members of the appropriate SAE task force.

The revision request form and instructions for updating the registers to this part of ISO 15031 can be obtained on the Registration Authority's website at:

http://www.sae.org/servlets/works/committeeHome.do?comtID=TEVDS14

The column titled "Resources" shows a document with the title: J1979-DA_Revision_Request_Form.doc. Double click on the name and you will be asked to download the document with the file name:

SAE_J1979-DA_Revision_Request_Form.doc

Fill out the revision request form with your request.

Please send an e-mail with the completed revision request form as an attachment to:

SAE Headquarters
755 West Big Beaver Road
Suite 1600
Troy, MI 48084-4093, USA
Fax: +1 (248) 273-2494
Email: saej1979@sae.org

Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics —

Part 5: Emissions-related diagnostic services

1 Scope

This part of ISO 15031 is intended to satisfy the data reporting requirements of On-Board Diagnostic (OBD) regulations in the United States and Europe and any other region that may adopt similar requirements in the future. This part of ISO 15031 specifies

a) message formats for request and response messages,

b) timing requirements between request messages from external test equipment and response messages from vehicles and between those messages and subsequent request messages,

c) behaviour of both the vehicle and external test equipment if data are not available, and

d) a set of diagnostic services, with corresponding content of request and response messages, to satisfy OBD regulations.

This part of ISO 15031 includes capabilities required to satisfy OBD requirements for multiple regions, model years, engine types, and vehicle types. Those regulations are not yet final for some regions and are expected to change in the future. This part of ISO 15031 makes no attempt to interpret the regulations and does not include applicability of the included diagnostic services and data parameters for various vehicle applications. The user of this part of ISO 15031 is responsible for verifying the applicability of each clause of this part of ISO 15031 for a specific vehicle, engine, model year, and region.

This part of ISO 15031 specifies diagnostic services and functionally addressed request/response messages required to be supported by motor vehicles and external test equipment for diagnostic purposes which pertain to motor vehicle emission-related data. Any external test equipment meeting the requirements of ISO 15031-4 use these messages to retrieve emissions-related information from the vehicle.

Each clause in this part of ISO 15031 which specifies additional details to existing clauses of ISO 9141-2, ISO 14230-4, SAE J1850, and ISO 15765-4 supersede those specifications.

This part of ISO 15031 references SAE J1979-DA (Digital Annex), which includes all definitions of PIDs, OBDMIDs, TIDs, and INFOTYPEs.

This part of ISO 15031 provides the mechanism to satisfy the requirements included in the country-specific regulations and not all capabilities included in this part of ISO 15031 are required by the country-specific regulations. This part of ISO 15031 is not considered a final authority for interpretation of the regulations. Therefore, readers should determine the applicability of capabilities defined in this part of ISO 15031 for their own specific needs.

2 Normative references

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

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

ISO 14229-2, *Road vehicles — Unified diagnostic services (UDS) — Part 2: Session layer services*

ISO 14230-2, *Road vehicles — Diagnostic communication over K-Line (DoK-Line) — Part 2: Data link layer*

ISO 14230-4:2000, *Road vehicles — Diagnostic systems — Keyword Protocol 2000 — Part 4: Requirements for emission-related systems*

ISO 15765-2, *Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) — Part 2: Transport protocol and network layer services*

ISO 15765-4, *Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) — Part 4: Requirements for emissions-related systems*

ISO 15031-2, *Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics — Part 2: Guidance on terms, definitions, abbreviations and acronyms*

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

SAE J1979-DA, *Digital Annex of E/E Diagnostic Test Modes*

SAE J2012-DA, *Digital Annex of Diagnostic Trouble Code Definitions and Failure Type Byte Definitions*

3 Terms, definitions, and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 14229-2, ISO 14230-2, ISO 15031-2, and ISO 15765-2 and the following apply.

3.1.1
absolute throttle position sensor
value intended to represent the throttle opening

Note 1 to entry: For systems where the output is proportional to the input voltage, this value is the percent of maximum input signal. For systems where the output is inversely proportional to the input voltage, this value is 100 % minus the percent of maximum input signal. Throttle position at idle usually indicates greater than 0 % and throttle position at wide open throttle usually indicates less than 100 %.

3.1.2
bank
specific group of cylinders sharing a common control sensor

Note 1 to entry: Bank 1 always contains cylinder number 1 and bank 2 the opposite bank.

Note 2 to entry: If there is only one bank, the DTCs for bank #1 DTCs are used and the word bank may be omitted. With a single “bank” system utilizing multiple sensors, bank #1 DTCs are used in identifying the sensors as #1, #2, and #3 in order as they move further away from the cylinder.

3.1.3
base fuel schedule
fuel calibration schedule programmed into the powertrain control module or PROM when manufactured or when updated by an off-board source, prior to any learned on-board correction

3.1.4
calculated load value
⟨spark ignition vehicles⟩ typically an indication of the current airflow divided by peak airflow at wide open throttle as a function of rpm, where airflow is corrected for altitude and ambient temperature

Note 1 to entry: Both spark ignition and compression ignition vehicles can use an alternate definition that substitutes engine torque in place of airflow in the calculation.

Note 2 to entry: This definition provides a number (without unit) and provides the service technician with an indication of the percent engine capacity that is being used.

3.1.5
client
function that is part of the tester and that makes use of the diagnostic services

Note 1 to entry: A tester normally makes use of other functions such as database management, specific interpretation, and man-machine interface.

3.1.6
continuous monitoring
sampling at a rate no fewer than two samples per second

Note 1 to entry: If, for control purposes, a computer input is sampled less frequently, the signal of the component may instead be evaluated each time sampling occurs.

3.1.7
convention
Cvt
column integrated in each message table which marks each parameter included

Note 1 to entry: The following conventions are used: C = Conditional: the parameter marked "C" in a request/response message is present only under a condition specified in the bottom row of the message table; M = Mandatory: the parameter marked "M" in a request/response message table is always present; U = User (optional): the parameter marked "U" in a request/response message table is supplied depending on dynamic usage by the manufacturer. The convention recommends a mnemonic, which might be used for implementation. In no case is the specified mnemonic ever a mandatory requirement for any implementation.

3.1.8
electronic control unit
ECU
generic term for any electronic control unit

3.1.9
emissions-related DTC
DTC which is set when a malfunction causes vehicle emissions to exceed legislated emission thresholds or is otherwise required to be set as specified by on-board diagnostics legislation (e.g. disables another part of the diagnostic system)

Note 1 to entry: Normally, the malfunction indicator (MI) is illuminated at the same time as the emissions-related DTC is set. The determination of which DTCs are emissions-related is made by the vehicle manufacturer for each vehicle, as specified by on-board diagnostic legislation.

3.1.10
fuel trim
FT
feedback adjustments to the base fuel schedule

Note 1 to entry: Short-term fuel trim refers to dynamic or instantaneous adjustments. Long-term fuel trim refers to much more gradual adjustments to the fuel calibration schedule than short-term trim adjustments. These long-term adjustments compensate for vehicle differences and gradual changes that occur over time.

3.1.11
negative numbers
signed binary, the most significant bit (MSB) of the binary number used to indicate positive (0)/ negative (1)

Note 1 to entry: 2's complement: negative numbers are represented by complementing the binary number and then adding 1.

EXAMPLE $-0{,}99 = 8001_{16} = 1000\ 0000\ 0000\ 0001_2$

$0 = 0000_{16} = 0000\ 0000\ 0000\ 0000_2$

$+0{,}99 = 7FFF_{16} = 0111\ 1111\ 1111\ 1111_2$

Note 2 to entry: (–0,99) + (+0,99) = 0.

3.1.12
number
expressed by this symbol "#"

3.1.13
P2, P3 timing parameter
application timing parameters for the ECU(s) and the external test equipment

3.1.14
$P2_{CAN\_min}$ timing parameter
CAN application timing parameter with the minimum value for the ECU(s) and the external test equipment to start the response message

3.1.15
$P2_{CAN\_max}$ timing parameter
CAN application timing parameter with the maximum value for the ECU(s) and the external test equipment to indicate a response message

3.1.16
$P2_{reload}$ timing parameter
CAN application timing parameter with the maximum value ($P2_{CAN\_max}$) for external test equipment only

3.1.17
server
function that is part of an ECU that provides the diagnostic services

Note 1 to entry: This part of ISO 15031 differentiates between the server, i.e. the function, and the electronic control unit so that it remains independent from the implementation.

3.1.18
service
information exchange initiated by a client (external test equipment) in order to require diagnostic information from a server (ECU) and/or to modify its behavior for diagnostic purposes

Note 1 to entry: This is also the equivalent of test mode or mode.

3.2 Abbreviated terms

.con confirmation

.ind indication

.req request

CRC	cyclic redundancy check
CVN	calibration verification number
DTC	diagnostic trouble code
ECM	engine control module
ERR	error detection byte
EWMA	exponential weighted moving average
FF	first frame
ISR	interrupt service routine
LSB	least significant bit
MI	malfunction indicator
MIL	malfunction indicator light
MSB	most significant bit
N_PDU	network protocol data unit
N/A	not applicable
NRC	negative response code
NVRAM	non-volatile memory
OBDMID OBD	monitor identifier
PID	parameter identifier
PCI	protocol control information
RSP	in-frame response
SF	single frame
SOM	start of message
T_AE	virtual transport interface address extension
T_Data [ ]	virtual transport interface data field
T_Mtype	virtual transport interface message type
T_Length	virtual transport interface length information
T_PDU	virtual transport interface protocol data unit
T_Result	virtual transport interface result
T_SA	virtual transport interface source address
T_TA	virtual transport interface target address
T_TAtype	virtual transport interface target address type

TCM	transmission control module
TID	test identifier
UASID	unit and scaling identifier
VIN	vehicle identification number

4 Conventions

This International Standard is based on the conventions specified in the OSI Service Conventions (ISO/IEC 10731) as they apply for diagnostic services.

5 Document overview

Figure 2 illustrates the emissions-related OBD in ISO 15765-4, SAE J1850, ISO 9141-2, and ISO 14230-4. The protocol initialization identifies whether ISO 15765-4 DoCAN or SAE J1850 or ISO 14230-4 DoK-Line or ISO 9141-2 is the data link layer supported by the vehicle. This International Standard references the standards as an applicable data link for emissions-related OBD.

This part of ISO 15031 specifies the applicable emissions-related diagnostic services. It specifies the data record structures and references SAE J1930-DA, SAE J1979-DA, and SAE J2012-DA which include all emissions-related OBD data definitions.

6 Technical requirements

6.1 General requirements

The requirements specified in this Clause are necessary to ensure proper operation of both the external test equipment and the vehicle during diagnostic procedures. External test equipment, when using the messages specified, shall not affect normal operation of the emission control system.

IMPORTANT — New emissions-related vehicle technology required the definition of new PIDs and INFOTYPEs. The data parameter set for several new definitions exceed the specified limit of message length for ISO 9141-2, ISO 14230-4, and SAE J1850 protocols. It is the vehicle manufacturer's responsibility to implement the ISO 15765-4 DoCAN protocol in order to achieve legislative compliance of the emissions-related OBD systems in the vehicle.

6.2 Diagnostic service requirements

6.2.1 Multiple responses to a single data request

The request messages are functional messages, which means that the external test equipment will request data without knowledge of which ECU(s) on the vehicle will respond. In some vehicles, multiple ECUs might respond with the information requested. Any external test equipment requesting information shall therefore have provisions for receiving multiple responses.

IMPORTANT — All emissions-related OBD ECUs, which at least support one of the services defined in this part of ISO 15031, shall support service 01_{16} and PID 00_{16}. Service 01_{16} with PID 00_{16} is defined as the universal "initialization/keep alive/ping" message for all emissions-related OBD ECUs.

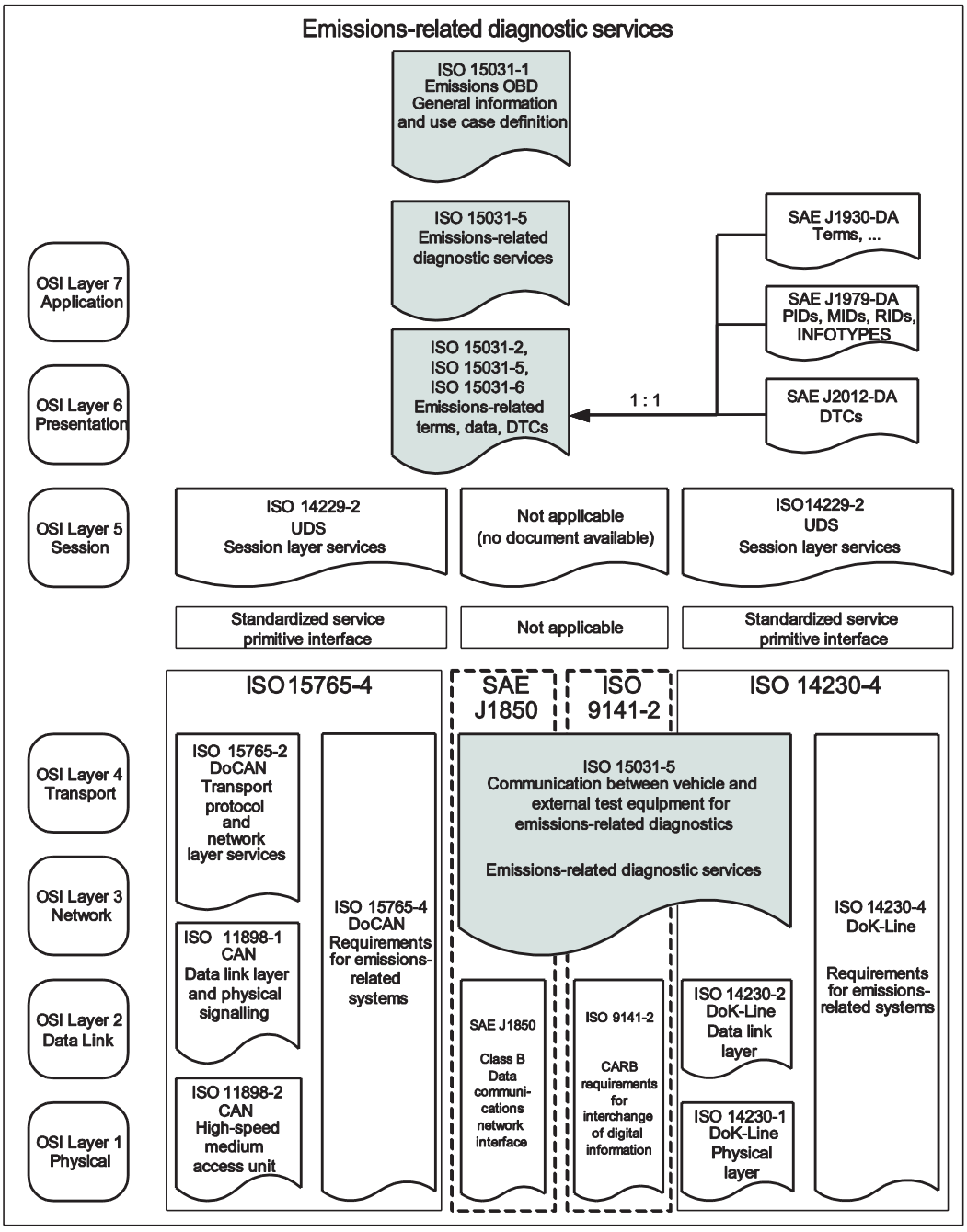


Figure 2 — Emissions-related OBD in ISO 15765-4, SAE J1850, ISO 9141-2, and ISO 14230-4 according to OSI model

6.2.2 Application timing parameter definition

6.2.2.1 Overview

The definition of P2 and P3 is included in this Clause. A subscript is added to each timing parameter to identify the following protocol:

— $P2_{K\text{-}line}$, $P3_{K\text{-}line}$: P2, P3 for ISO 9141-2 and ISO 14230-4 protocols;

— $P2_{J1850}$: P2 for SAE J1850 protocol;

— $P2_{CAN}$: P2 for ISO 15765-4 protocol.

IMPORTANT — The vehicle manufacturer is responsible for specifying a shorter P2 timing window than specified in this part of ISO 15031 for each emission-related server/ECU in the vehicle in order to make sure that network topology delays of the vehicle architecture are considered.

6.2.2.2 Definition for ISO 9141-2

For ISO 9141-2 interfaces, data link layer response time requirements (P1, P4) are specified in ISO 9141-2.

Table 2 specifies the application timing parameter values for P2 and P3.

Table 2 — Definition of ISO 9141-2 application timing parameter values

Parameter	Minimum value ms	Maximum value ms	Description
$P2_{K\text{-}line}$ Key Bytes: 08_{16} 08_{16}	25	50	Time between external test equipment request message and the transmission of the ECU(s) response message(s). Each OBD ECU shall start sending its response message within $P2_{K\text{-}line}$ after the request message has been correctly received. Subsequent response messages shall also be transmitted within $P2_{K\text{-}line}$ of the previous response message for multiple message responses.
$P2_{K\text{-}line}$ Key Bytes: 94_{16} 94_{16}	0	50	Time between external test equipment request message and the transmission of the ECU response message(s). The OBD ECU shall start sending its response message within $P2_{K\text{-}line}$ after the request message has been correctly received. Subsequent response messages shall also be transmitted within $P2_{K\text{-}line}$ of the previous response message for multiple message responses.
$P3_{K\text{-}line}$	55	5 000	Time between the end of an ECU(s) successful transmission of response message(s) and start of new external test equipment request message. The external test equipment may send a new request message if all response messages related to the previously sent request message have been received and if $P3_{K\text{-}line}$ minimum time has expired. ECU implementation guideline: TX (transmit) and RX (receive) lines are connected. Each transmitted byte is read back by the receiver in the ECU. Upon the reception of a received byte, e.g. last byte of a request message (checksum) from the tester, the ECU shall reset the P3 timer value to zero. If the ECU supports the request message, it will start transmitting the response message within the P2 timing window. Each transmitted byte will cause the P3 timer value to be reset. If the ECU does not support the request and does not send a response message, then in a single OBD ECU system, the P3 is started after the last byte received of the request message. In a multiple OBD ECU system, a response message by one or more ECUs shall cause the P3 timer value to be reset in all ECUs including any ECU not supporting the request message.

6.2.2.3 Definition for ISO 14230-4

For ISO 14230-4 interfaces, data link layer response time requirements are specified in ISO 14230-4.

Table 3 specifies the application timing parameter values for P2 and P3.

Table 3 — Definitions of ISO 14230-4 application timing parameter values

Parameter	Minimum value ms	Maximum value ms	Description
$P2_{K\text{-}line}$	25	50	Time between external test equipment request message and the transmission of the ECU(s) response message(s). Each OBD ECU shall start sending its response message within $P2_{K\text{-}line}$ after the request message has been correctly received. Subsequent response messages shall also be transmitted within $P2_{K\text{-}line}$ of the previous response message for multiple message responses.
$P3_{K\text{-}line}$	55	5 000	Time between the end of an ECU(s) successful transmission of response message(s) and start of new external test equipment request message. The external test equipment may send a new request message if all response messages related to the previously sent request message have been received and if $P3_{K\text{-}line}$ minimum time has expired. ECU implementation guideline: TX (transmit) and RX (receive) line are connected. Each transmitted byte is read back by the receiver in the ECU. Upon the reception of a received byte, e.g. last byte of a request message (checksum) from the tester, the ECU shall reset the P3 timer value to zero. If the ECU supports the request message, it will start transmitting the response message within the P2 timing window. Each transmitted byte will cause the P3 timer value to be reset. If the ECU does not support the request and does not send a response message, then in a single OBD ECU system, the P3 is started with the last byte received of the request message. In a multiple OBD ECU system, a response message by any one or more ECUs shall cause the P3 timer value to be reset in all ECUs including any ECU not supporting the request message.

6.2.2.4 Data link layer interface adaptation

6.2.2.4.1 General information

This part of ISO 15031 makes use of the data link layer services defined in ISO 14230-2 for the transmission and reception of diagnostic messages. This section defines the mapping of the virtual data link PDU (T_PDU) in ISO 14229-2 onto the K-Line data link layer PDU (DL_PDU) in ISO 14230-2.

NOTE The data link layer services are used to perform the application layer and diagnostic session management timing.

6.2.2.4.2 Mapping of data link independent service primitives onto K-Line data link dependent service primitives

Table 4 specifies the mapping interface between the ISO 14230-2 DoK-Line Part 2: Data link layer services and the ISO 14229-2 UDS Part 2.

Table 4 — Mapping of T_PDU service primitives onto DL_PDU service primitives

Transport/network layer service primitives (data link independent according to ISO 14229-2)	DoK-Line data link layer service primitives (data link dependent according to ISO 14230-2)
T_Data.indication	DL_Data.indication
T_DataSOM.indication	DL_DataFB.indication
T_Data.confirm	DL_Data.confirm
T_Data.request	DL_Data.request

6.2.2.4.3 Mapping of T_PDU onto DL_PDU for message transmission

The parameters of the application layer protocol data unit defined to request the transmission of a diagnostic service request/response are mapped in accordance with Table 5 onto the parameters of the data link layer protocol data unit for the transmission of a message in the client/server.

Table 5 — Mapping of T_PDU parameter onto DL_PDU parameter

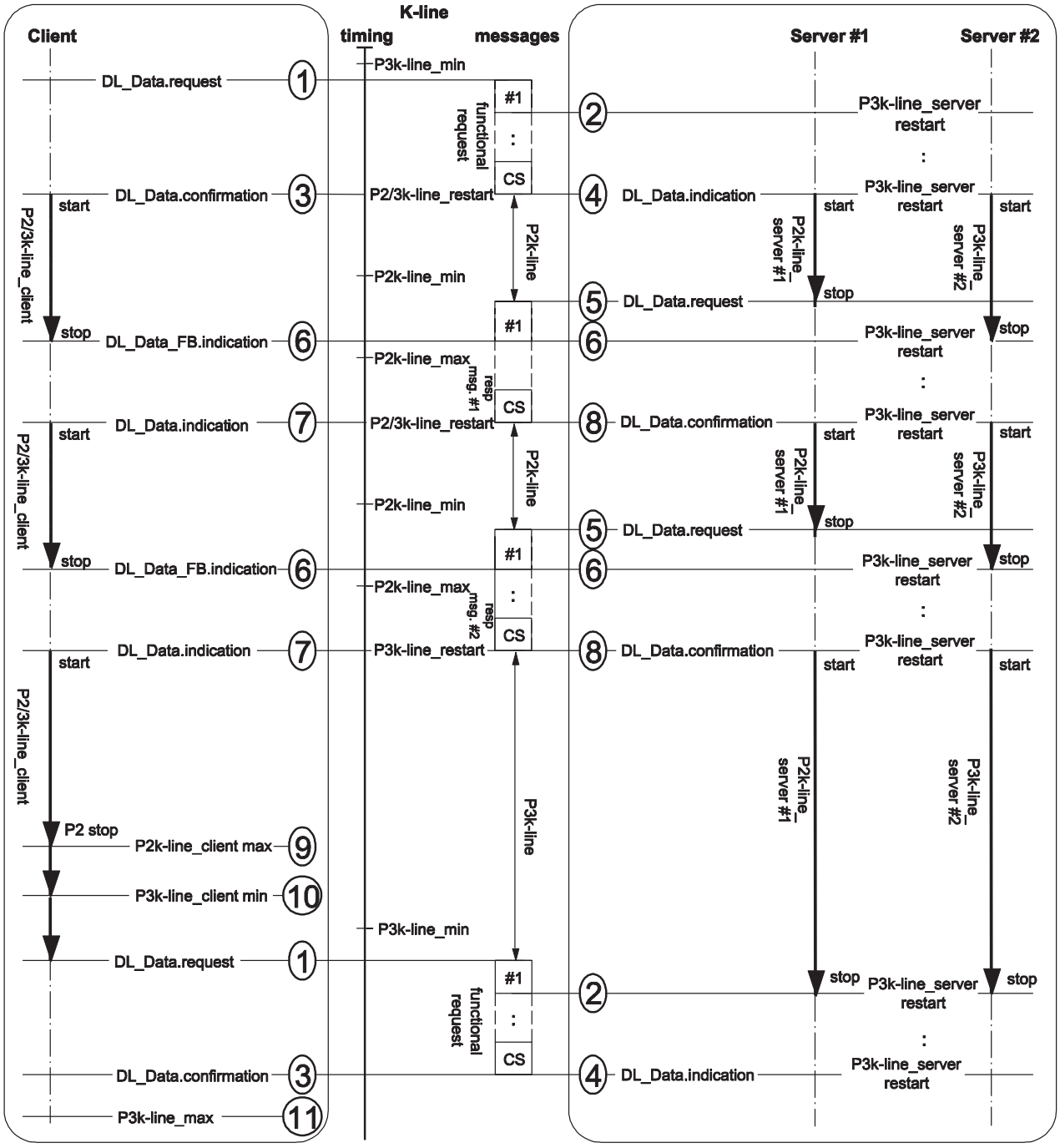
T_PDU parameter (data link independent according to ISO 14229-2)	DL_PDU parameter (DoK-Line data link dependent according to ISO 14230-2)
T_Mtype	N/A (always set to “diagnostics”)
T_SA	DL_SA
T_TA	DL_TA
T_TAtype	DL_TAtype
T_AE	N/A
T_Data [ ]	< MessageData >
T_Length	< Length >
T_Result	< DL_Result >

6.2.2.5 Implementation guidance example for ISO 9141-2 and ISO 14230-4 protocols

This subclause provides an implementation example for client/external test equipment and server/ECU. It is assumed that the client (external test equipment) communicates to a vehicle with two (2) emission-related OBD servers (ECUs). The client requests a CVN, which is only supported by server #1 (ECU#1) with two (2) response messages. Server #2 (ECU#2) is not flash programmable. Figure 3 graphically depicts the timing handling in the client and two (2) servers for a functionally addressed request message. A description following Figure 3 references the points marked in the figure.

From a server point of view, there is no difference in the timing handling compared to a physically addressed request message. The server shall reset the $P3_{K\text{-}line}$ timer value on each received byte regardless of whether the byte is part of a request message or a response message from any other server or an echo from its transmit line. There are several methods in which a server can implement the timing handling. The implementation of timing parameters is not part of this part of ISO 15031 but has an important system supplier responsibility. Some general server timing parameter implementation guidelines are described in this subclause. The server time stamps each receiver interrupt event and restarts/resets the $P3_{K\text{-}line\_server}$ timer or timing value, e.g. ISR time stamps received byte, and processing of the received information is performed outside the ISR. For simplification of the diagram, Figure 3 only shows a $P3_{K\text{-}line\_server}$ restart after the reception of the first byte and last byte (checksum) of a received message. The $P3_{K\text{-}line\_server}$ restart is required on each received byte. The received message can be either a request message from the client or a response message from any other server connected and initialized by the 33_{16} address. If the server has received a complete message, it compares the target address with the 33_{16} address.

Figure 3 shows the client and two (2) initialized servers connected via K-line (either ISO 9141-2 or ISO 14230-4 protocol). The relevant events for the client and both servers are marked and described.



Key

1 The diagnostic application of the client starts the transmission of a functionally addressed request message by issuing a T_Data.request to its data link layer. The data link layer transmits the request message to the servers.

2 Both servers and the client receive a byte of a message via a receive interrupt by the UART. The ISR (interrupt service routine) either restarts the $P2_{K\text{-line}}$/$P3_{K\text{-line}}$ timers or time stamps the received byte.

3 The completion of the request message is indicated in the client with T_Data.confirmation. When receiving the T_Data.confirmation, the client starts its $P2_{K\text{-line}}$ and $P3_{K\text{-line}}$ timer using the default reload values $P2_{K\text{-line\_max}}$ and $P3_{K\text{-line\_max}}$.

4 If the last message byte is received, each server checks whether the received message includes a target address which matches the 33_{16} address. If the result is a match (server #1 and #2), then the completion of the request message is indicated in the servers via T_Data.indication and each server determines whether it supports the request and has a message available to respond with. If a server determines that the address in the received message is different from 33_{16} or if the address is a match but no response needs to be sent (server #2), the P2 timer is stopped. Since the $P3_{K\text{-}line}$ timer has already been restarted, no further action is required. If a response message is available and has to be sent (server #1, but not server #2), then the transmission of the response message shall be started after $P2_{K\text{-}line\_min}$ timing is expired.

5 Server #1 starts the response message by indicating a T_Data.request from the application to the data link layer and at the same time stops its $P2_{K\text{-}line}$ timer.

6 Both servers and the client receive a byte of a message via a receive interrupt by the UART. The ISR (interrupt service routine) restarts the $P2_{K\text{-}line}$/$P3_{K\text{-}line}$ timers or time stamps the received byte and the client issues a T_Data_FB.indication to the application layer.

7 The completion of the response message is indicated in the client with T_Data.indication. When receiving the T_Data.indication, the client starts its $P2_{K\text{-}line}$ and $P3_{K\text{-}line}$ timer using the default reload values $P2_{K\text{-}line\_max}$ and $P3_{K\text{-}line\_max}$.

8 Both servers have received the last byte of a message via a receive interrupt by the UART. The ISR (interrupt service routine) either resets the $P2_{K\text{-}line}$/$P3_{K\text{-}line}$ timers or time stamps the received byte. The completion of the response message (e.g. length and checksum check) is indicated in server #1 via T_Data.confirmation. If server #1 does not want to send further response messages, it stops its P2 timer. In server #2, the message is received and the $P3_{K\text{-}line}$ timer is restarted, but no T_Data.indication is forwarded to the application because the target address does not match the 33_{16} (target address of this message is the tester address $F1_{16}$).

9 The client application detects a $P2_{K\text{-}line\_max}$ timeout, which indicates that all response messages from all servers are received.

10 The client application indicates that $P3_{K\text{-}line\_min}$ is reached and that the $P3_{K\text{-}line}$ timing window is now open to send a new request message (see 1).

11 $P3_{K\text{-}line\_max}$ timeout indicates that the client is required to start a new initialization prior to sending a new request message.

Figure 3 — ISO 9141-2 and ISO 14230-4 protocol client and server timing behaviour

6.2.2.6 Definition for SAE J1850

For SAE J1850 network interfaces, the on-board systems shall respond to a request within $P2_{J1850}$ of a request or a previous response message. With multiple response messages possible from a single request message, this allows as much time as is necessary for all ECUs to access the data link and transmit their response message(s). If there is no response message within this time period, the external test equipment can either assume no response message will be received, or if a response message has already been received, that no more response messages will be received. The application timing parameter value $P2_{J1850}$ is specified in Table 6.

Table 6 — Definition of SAE J1850 application timing parameter values

Parameter	Minimum value ms	Maximum value ms	Description
$P2_{J1850}$	0	100	Time between external test equipment request message and the successful transmission of the ECU(s) response message(s). Each OBD ECU shall attempt to send its response message (or at least the first of multiple response messages) within $P2_{J1850}$ after the request message has been correctly received. Subsequent response messages shall also be transmitted within $P2_{J1850}$ of the previous response message for multiple message responses.

6.2.2.7 Definition for ISO 15765-4

For CAN bus systems based on ISO 15765-4, the (all) responding ECU(s) of the on-board system shall start the response message to a request message within $P2_{CAN}$. Table 7 specifies the application timing parameter values for P2.

Table 7 — Definition of ISO 15765-4 application timing parameter values

Parameter	Minimum value $P2_{CAN\_min}$ ms	Maximum value $P2_{CAN\_max}$ ms	Description
$P2_{CAN}$	0	50	This is a system-wide parameter related to diagnostic response times. Each server (ECU) is required to respond to a request between $P2_{CAN\_min}$ and $P2_{CAN\_max}$. A client (tester) shall wait for at least $P2_{CAN\_max}$ for the single-frame (SF) or first-frame (FF) of a response. $P2_{CAN}$ is the time until the first indication of a multiple-frame response message (FirstFrame). The client shall not process the response until the complete message (last ConsecutiveFrame) has been received. For clients (testers) which also support UDSonCAN for enhanced diagnostics, a $P2_{reload}$ mechanism is required. Upon receiving the SF or FF, the client (tester) shall reload its $P2_{CAN}$ timer with a value of at least $P2_{CAN\_max}$ and restart the timer. Once the client's (tester's) $P2_{CAN}$ timer expires without receiving a SF or FF, the client (tester) may assume no more responses are forthcoming.
$P2^*_{CAN}$	0	5 000	Time between the successful reception of a negative response message with NRC 78_{16} and the next response message (positive or negative message). See Table 11 for a list of which services support the use of NRC 78_{16}.

NOTE The network layer timing parameters for the multiple-frame response are not shown. Network layer timing requirements for legislated diagnostic messages are specified in ISO 15765-4.

6.2.2.8 Transport/Network layer interface adaptation

6.2.2.8.1 General information

This part of ISO 15031 makes use of the network layer services defined in ISO 15765-2 for the transmission and reception of diagnostic messages. This section defines the mapping of the virtual data link PDU (T_PDU) onto the independent transport/network layer protocol data units of the CAN data link specific network layer (N_PDU).

NOTE The transport/network layer services are used to perform the application layer and diagnostic session management timing.

6.2.2.8.2 Mapping of data link independent service primitives onto CAN data link dependent service primitives

Table 8 specifies the mapping interface between the ISO 15765-2 DoCAN Part 2 and the ISO 14229-2 UDS Part 2.

Table 8 — Mapping of T_PDU service primitives onto N_PDU service primitives

Transport/network layer service primitives (data link independent according to ISO 14229-2)	DoCAN network layer service primitives (data link dependent according to ISO 15765-2)
T_Data.indication	N_USData.indication
T_DataSOM.indication	N_USDataFF.indication
T_Data.confirm	N_USData.confirm
T_Data.request	N_USData.request

6.2.2.8.3 Mapping of T_PDU onto N_PDU for message transmission

The parameters of the application layer protocol data unit defined to request the transmission of a diagnostic service request/response are mapped in accordance with Table 9 onto the parameters of the network layer protocol data unit for the transmission of a message in the client/server.

The network layer confirmation of the successful transmission of the message (N_USData.con) is forwarded to the application because it is needed in the application for starting those actions, which shall be executed immediately after the transmission of the request/response message (ECUReset, BaudrateChange, etc.).

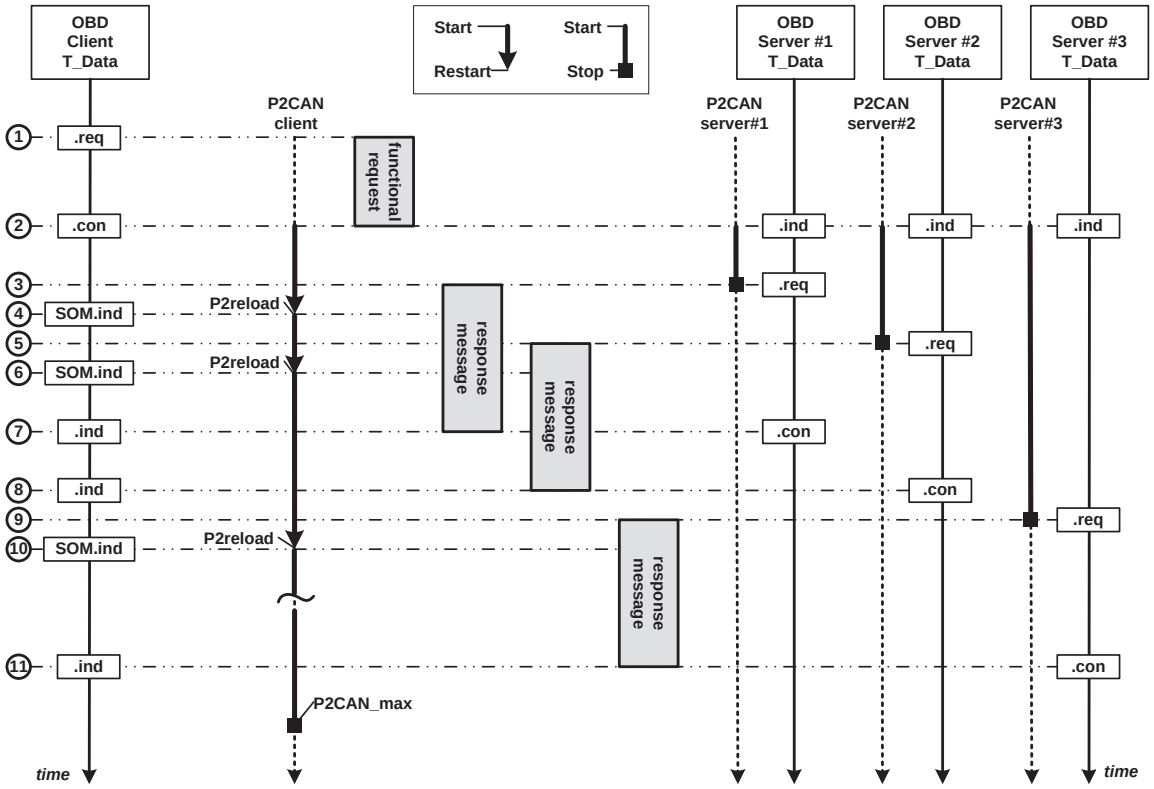
Table 9 — Mapping of T_PDU parameter onto N_PDU parameter

T_PDU parameter (data link independent according to ISO 14229-2)	N_PDU parameter (CAN data link dependent according to ISO 15765-2)
T_Mtype	N_Mtype
T_SA	N_SA
T_TA	N_TA
T_TAtype	N_TAtype
T_AE	N_AE
T_Data [ ]	<MessageData>
T_Length	<Length>
T_Result	<N_Result>

6.2.2.9 Implementation guidance example for ISO 15765-4 protocol

6.2.2.9.1 Functional OBD communication during default session

Figure 4 graphically depicts the timing handling in the client and three servers for a functionally addressed request message during the default session. A description following Figure 4 references the points marked in the figure.



Key

1 Client T_Data.req: diagnostic application issues functionally addressed request message to network layer.

2 All server T_Data.ind: network layer issues to diagnostic application the reception of a request message. All servers start the $P2_{CAN}$ timer using the value of $P2_{CAN} = P2_{CAN\_max}$.

Client T_Data.con: network layer issues to diagnostic application the confirmation of the completion of the request message. Client starts its $P2_{CAN\_Client}$ timer using the default reload value $P2_{CAN} = P2_{CAN\_max}$.

3 Server #1 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$. The response message can be a multi-frame or single-frame response message.

4 Client T_DataSOM.ind: network layer issues to diagnostic application the reception of a StartOfMessage which is initiated by the reception of a FirstFrame indication on CAN (see ISO 15765-2). Reload $P2_{CAN}$ with $P2_{CAN\_max}$ value.

5 Server #2 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$. The response message can be a multi-frame or single-frame response message

6 Client T_DataSOM.ind: network layer issues to diagnostic application the reception of a StartOfMessage which is initiated by the reception of a FirstFrame indication on CAN (see ISO 15765-2). Reload $P2_{CAN}$ with $P2_{CAN\_max}$ value.

7 Server #1 T_Data.con: network layer issues to diagnostic application the completion of the response message.
Client T_Data.ind: network layer issues to diagnostic application the completion of the response message.

8 Server #2 T_Data.con: network layer issues to diagnostic application the completion of the response message.
Client T_Data.ind: network layer issues to diagnostic application the completion of the response message.

9 Server #3 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$. The response message can be a multi-frame or single-frame response message.

10 Client T_DataSOM.ind: network layer issues to diagnostic application the reception of a StartOfMessage which is initiated by the reception of a FirstFrame indication on CAN (see ISO 15765-2). Reload $P2_{CAN}$ with $P2_{CAN\_max}$ value.

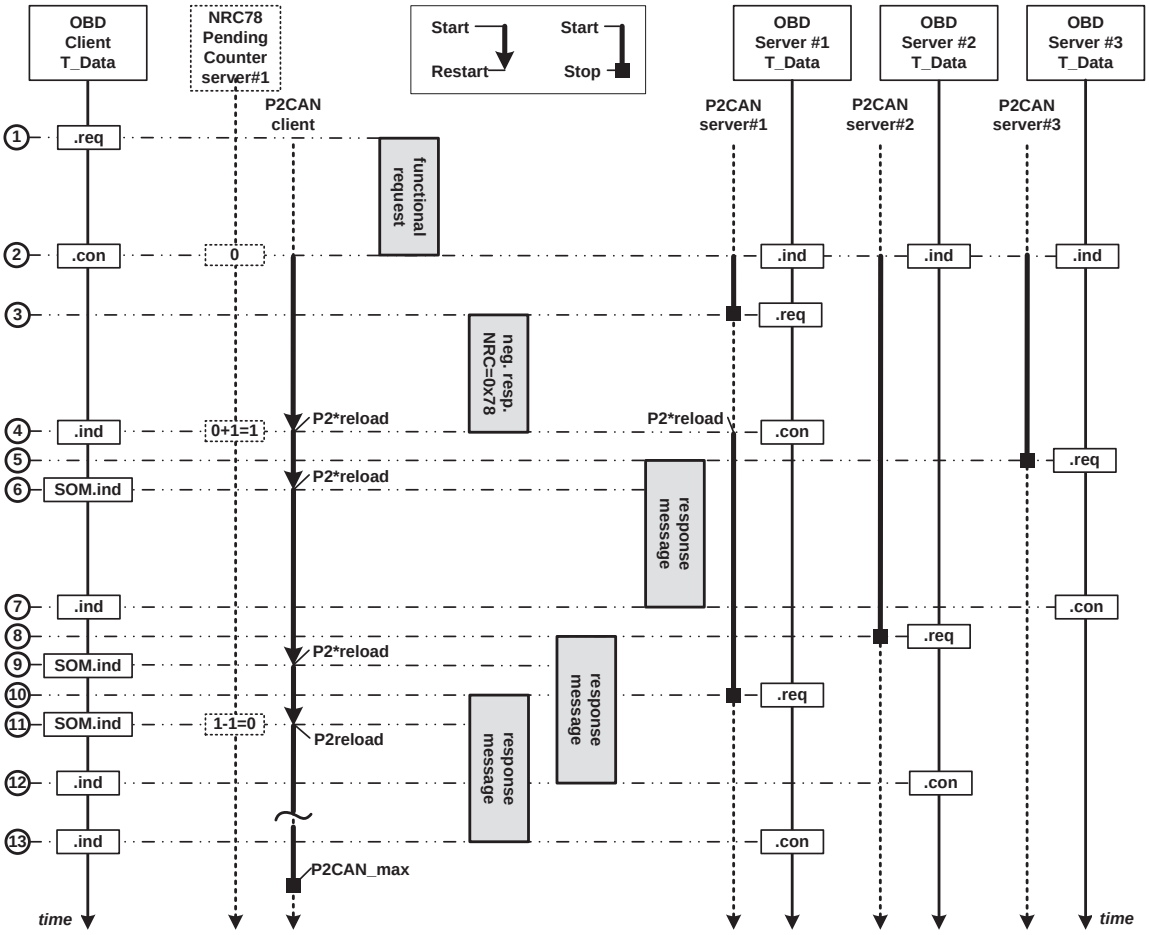
11 Server #2 T_Data.con: network layer issues to diagnostic application the completion of the response message.

Client T_Data.ind: network layer issues to diagnostic application the completion of the response message.

Figure 4 — Functional OBD communication — Default response timing

6.2.2.9.2 Functional OBD communication with enhanced response timing

Figure 5 illustrates the timing handling in the client and three (3) servers for a functionally addressed request message during the default session, where one server requests an enhanced response timing via a negative response message including NRC 78_{16}. A description following Figure 5 references the points marked in the figure.



Key

1 Client T_Data.req: diagnostic application issues functionally addressed request message to network layer.

2 All server T_Data.ind: network layer issues to diagnostic application the reception of a request message. All servers start the $P2_{CAN}$ timer using the value of $P2_{CAN}$ = P2 $_{CAN_}$max.

Client T_Data.con: network layer issues to diagnostic application the confirmation of the completion of the request message. Client starts its $P2_{CAN}$ timer using the default reload value $P2_{CAN}$ = $P2_{CAN\_max}$. All NRCPendingCounter = 0.

3 Server #1 T_Data.req: diagnostic application does not have the positive response message ready and issues negative response message with NRC = 78_{16} by a T_Data.req to the network layer within $P2_{CAN}$.

4 Server #1 T_Data.con: network layer issues to diagnostic application the completion of the response message.

Client T_Data.ind: network layer issues to diagnostic application the reception of a message. Since the received response message is a negative response message with NRC = 78_{16}, the NRCPendingCounter server #1 is incremented by 1 (0+1 = 1).

Reload $P2_{CAN}$ with $P2^*_{CAN\_max}$ value. Server#1 reloads $P2_{CAN}$ with $P2^*_{CAN\_max}$ value.

5 Server #3 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$.

6 Client T_DataSOM.ind: network layer issues to diagnostic application the reception of a StartOfMessage which is initiated by the reception of a FirstFrame indication on CAN (see ISO 15765-2). Reload $P2_{CAN}$ with $P2^*_{CAN\_max}$ value.

7 Server #3 T_Data.con: network layer issues to diagnostic application the completion of the response message.

Client T_Data.ind: network layer issues to diagnostic application the completion of the response message.

8 Server #2 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$.

9 Client T_DataSOM.ind: network layer issues to diagnostic application the reception of a StartOfMessage which is initiated by the reception of a FirstFrame indication on CAN (see ISO 15765-2). Client reloads $P2_{CAN}$ with $P2^*_{CAN\_max}$ value.

10 Server #1 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$

11 Client T_DataSOM.ind: network layer issues to diagnostic application the reception of a StartOfMessage which is initiated by the reception of a FirstFrame indication on CAN (see ISO 15765-2). Since the received response message is a positive response message, the NRCPendingCounter server #1 is decremented by 1 (1-1 = 0). Client reloads $P2_{CAN}$ with $P2_{CAN\_max}$ value.

12 Server #2 T_Data.con: network layer issues to diagnostic application the completion of the response message.
Client T_Data.ind: network layer issues to diagnostic application the completion of the response message.

13 Server #1 T_Data.con: network layer issues to diagnostic application the completion of the response message
Client T_Data.ind: network layer issues to diagnostic application the completion of the response message

Figure 5 — Functional OBD communication — Enhanced response timing

6.2.3 Minimum time between requests from external test equipment

6.2.3.1 ISO 9141-2, ISO 14230-4 — Minimum time between requests from external test equipment

For ISO 9141-2 (K-line) interfaces, the required times between request messages are specified in ISO 9141-2.

For ISO 14230-4 (K-line) interfaces, the required times between request messages are specified in ISO 14230-4. Figure 6 shows an example of a request message followed by four (4) response messages and another request message.

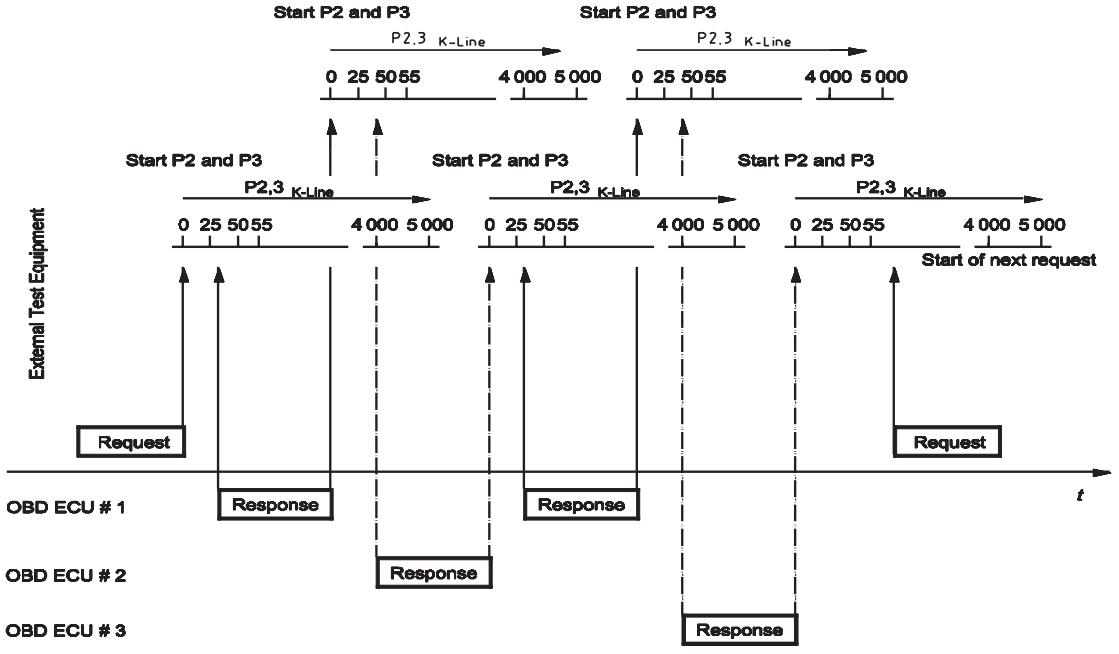


Figure 6 — ISO 9141-2 (Key bytes: 08_{16} 08_{16}) and ISO 14230-4 application timing parameter overview

6.2.3.2 SAE J1850 — Minimum time between requests from external test equipment

For SAE J1850 network interfaces, external test equipment shall always wait for a response message from the previous request or "no response" time-out before sending another request message. If the number of response messages is known and all response messages have been received, then the external test equipment is permitted to send the next request message immediately. If the number of

response messages is not known, then the external test equipment shall wait at least $P2_{J1850}$ maximum time.

Figure 7 illustrates an example of a request message followed by four (4) response messages and another request message.

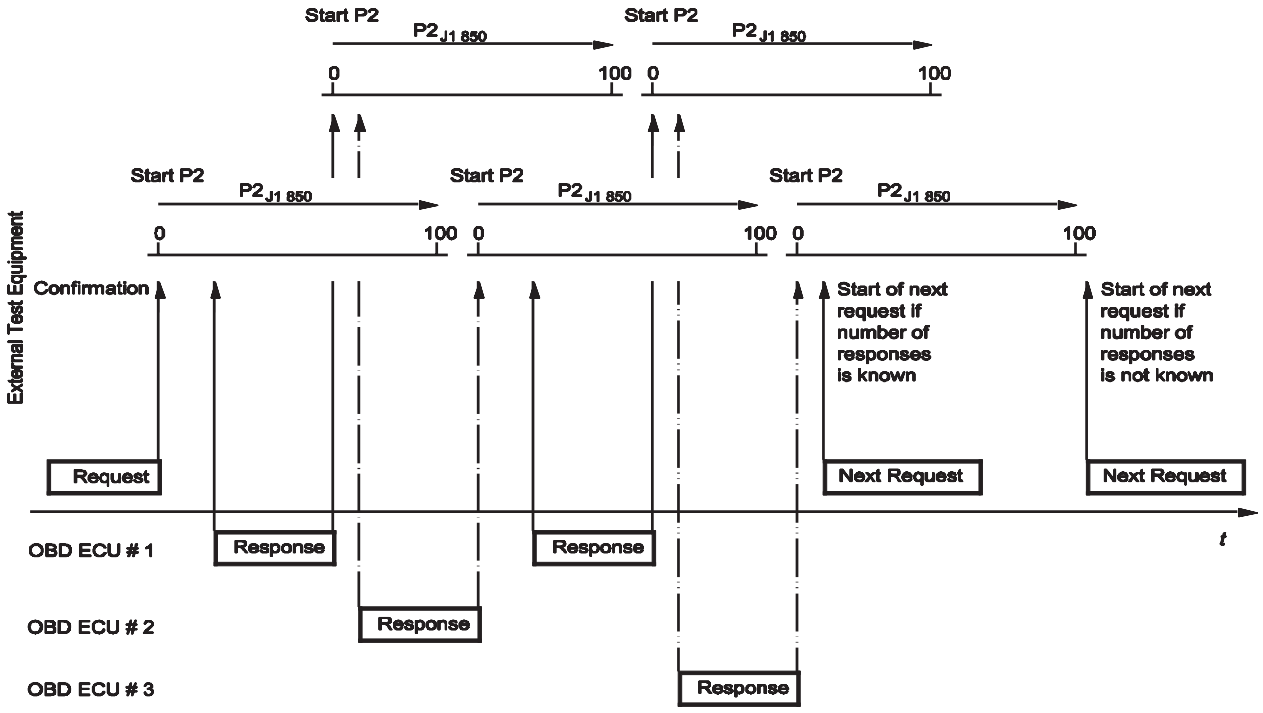


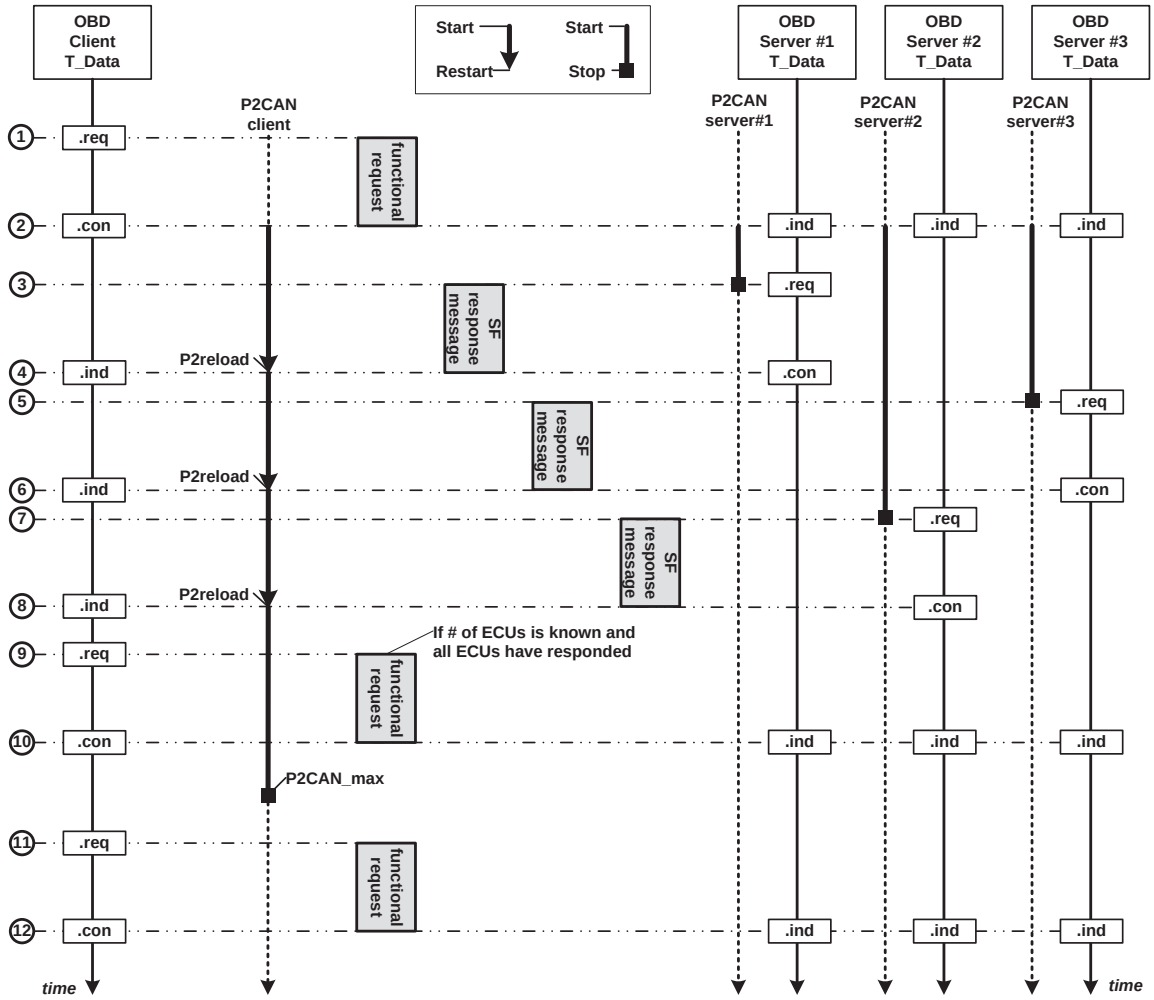
Figure 7 — SAE J1850 application timing parameter overview

6.2.3.3 ISO 15765-4 — Minimum time between requests from external test equipment

For ISO 15765-4 network interfaces, the external test equipment may send a new request message immediately after it has determined that all responses related to the previously sent request message have been received. If the external test equipment does not know whether it has received all response messages, (e.g. after sending the initial OBD request message: Service 01_{16}, PID 00_{16}), it shall wait until $P2_{CAN\_max}$ expires before sending another request. The timer $P2_{CAN}$ of the external test equipment starts with the confirmation of a successful transmission of the request message.

Figure 8 illustrates an example of a request message followed by three (3) single-frame response messages and another request message.

IMPORTANT — The $P2_{CAN\_reload}$ is performed by the client to identify whether more emissions-related OBD ECUs will send a response message. The $P2_{CAN\_reload}$ is not defined to check whether the entire response message is sent within $P2_{CAN\_max}$ timing.



Key

1 Client T_Data.req: diagnostic application issues functionally addressed request message to network layer.

2 All server T_Data.ind: network layer issues to diagnostic application the reception of a request message. All servers start the $P2_{CAN}$ timer using the value of $P2_{CAN} = P2_{CAN\_max}$.

Client T_Data.con: network layer issues to diagnostic application the confirmation of the completion of the request message. Client starts its $P2_{CAN}$ timer using the default reload value $P2_{CAN} = P2_{CAN\_max}$.

3 Server #1 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$. The response message is a single-frame message.

4 Server #1 T_Data.con: network layer issues to diagnostic application the completion of the response message. Client T_Data.ind: network layer issues to diagnostic application the completion of the response message. Client reloads $P2_{CAN}$ with $P2_{CAN\_max}$ value.

5 Server #3 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$. The response message is a single-frame message.

6 Server #3 T_Data.con: network layer issues to diagnostic application the completion of the response message. Client T_Data.ind: network layer issues to diagnostic application the completion of the response message. Client reloads $P2_{CAN}$ with $P2_{CAN\_max}$ value.

7 Server #2 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$. The response message is a single-frame message.

8 Server #2 T_Data.con: network layer issues to diagnostic application the completion of the response message.

Client T_Data.ind: network layer issues to diagnostic application the completion of the response message. Client reloads $P2_{CAN}$ with $P2_{CAN\_max}$ value

9 Client T_Data.req: diagnostic application issues functionally addressed request message to network layer. The client knows the number of ECUs which shall have responded to the previous request. Since all response messages have been received, the client is already allowed to issue a new functional request message

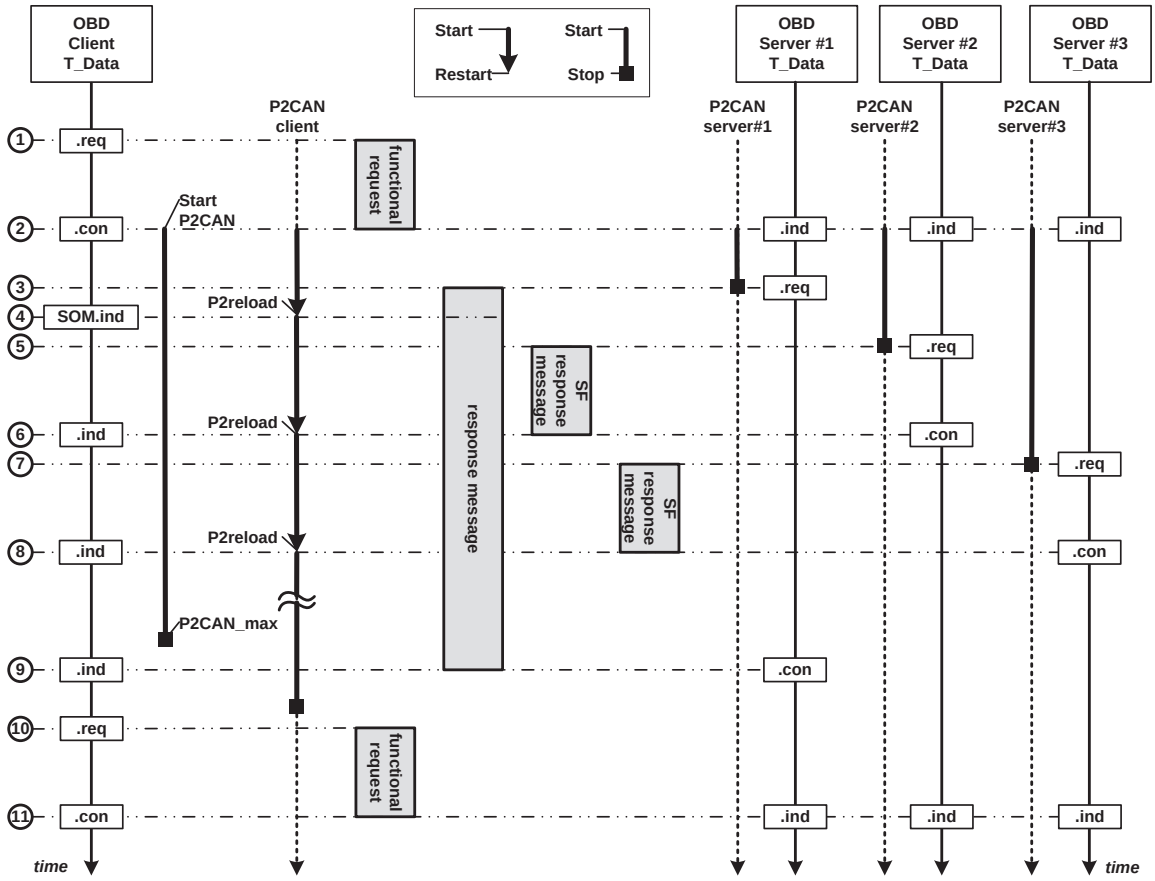
10 All server T_Data.ind: network layer issues to diagnostic application the reception of a request message. All servers start the $P2_{CAN}$ timer using the value of $P2_{CAN} = P2_{CAN\_max}$.
Client T_Data.con: network layer issues to diagnostic application the confirmation of the completion of the request message. Client starts its $P2_{CAN}$ timer using the default reload value $P2_{CAN} = P2_{CAN\_max}$ (not shown in figure).

11 Client T_Data.req: diagnostic application issues functionally addressed request message to network layer. The client does not know the number of ECUs which shall have responded to the previous request. Therefore, the client shall wait until $P2_{CAN} = P2_{CAN\_max}$ before it issues a new functional request message.

12 All server T_Data.ind: network layer issues to diagnostic application the reception of a request message. All servers start the $P2_{CAN}$ timer using the value of $P2_{CAN} = P2_{CAN\_max}$ (not shown in figure).
Client T_Data.con: network layer issues to diagnostic application the confirmation of the completion of the request message. Client starts its $P2_{CAN}$ timer using the default reload value $P2_{CAN} = P2_{CAN\_max}$ (not shown in figure).

Figure 8 — ISO 15765-4 application timing parameter (single-frame response messages) overview

Figure 9 illustrates an example of a request message followed by one (1) multiple-frame response message and two (2) single frames and another request message. The next request message can be sent immediately by the external test equipment after completion of all response messages in case the transmission of the response messages takes longer than $P2_{CAN\_max}$, even if the external test equipment does not know the number of responding ECUs.



Key

1 Client T_Data.req: diagnostic application issues functionally addressed request message to network layer.

2 All server T_Data.ind: network layer issues to diagnostic application the reception of a request message. All servers start the $P2_{CAN}$ timer using the value of $P2_{CAN} = P2_{CAN\_max}$.
Client T_Data.con: network layer issues to diagnostic application the confirmation of the completion of the request message. Client starts its $P2_{CAN}$ timer using the default reload value $P2_{CAN} = P2_{CAN\_max}$.

3 Server #1 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$.

4 Client T_DataSOM.ind: network layer issues to diagnostic application the reception of a StartOfMessage which is initiated by the reception of a FirstFrame indication on CAN (see ISO 15765-2). Client reloads $P2_{CAN}$ with $P2_{CAN\_max}$ value.

5 Server #2 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$. The response message is a single-frame message.

6 Server #2 T_Data.con: network layer issues to diagnostic application the completion of the response message.

Client T_Data.ind: network layer issues to diagnostic application the completion of the response message. Client reloads $P2_{CAN}$ with $P2_{CAN\_max}$ value.

7 Server #3 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$. The response message is a single-frame message.

8 Server #3 T_Data.con: network layer issues to diagnostic application the completion of the response message.

Client T_Data.ind: network layer issues to diagnostic application the completion of the response message. Client reloads $P2_{CAN}$ with $P2_{CAN\_max}$ value.

9 Server #1 T_Data.con: network layer issues to diagnostic application the completion of the response message.

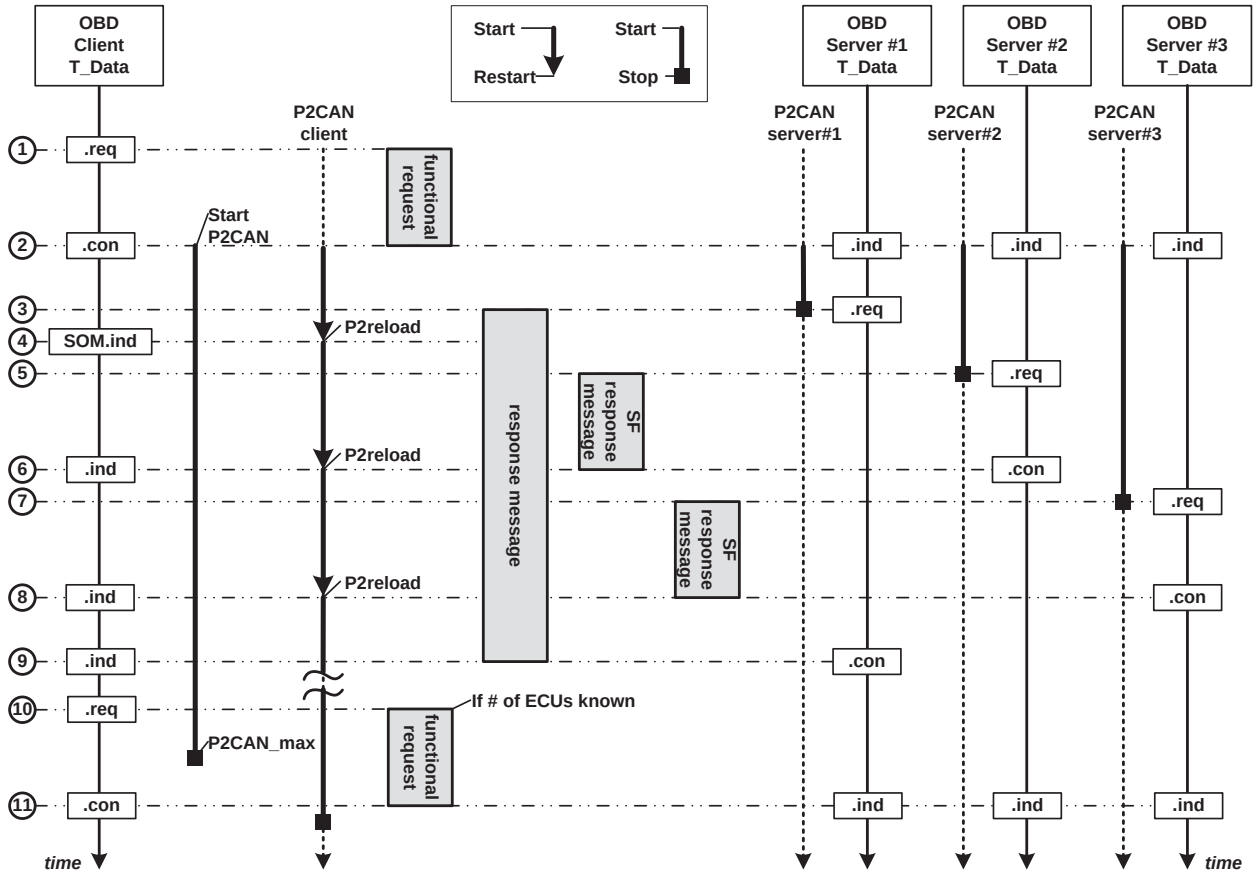
Client T_Data.ind: network layer issues to diagnostic application the completion of the response message.

10 Client T_Data.req: diagnostic application issues functionally addressed request message to network layer. The client does not know the number of ECUs which shall have responded to the previous request. Therefore, the client shall wait until $P2_{CAN} = P2_{CAN\_max}$ before it issues a new functional request message.

11 All server T_Data.ind: network layer issues to diagnostic application the reception of a request message. All servers start the $P2_{CAN}$ timer using the value of $P2_{CAN} = P2_{CAN\_max}$.
Client T_Data.con: network layer issues to diagnostic application the confirmation of the completion of the request message. Client starts its $P2_{CAN}$ timer using the default reload value $P2_{CAN} = P2_{CAN\_max}$.

Figure 9 — ISO 15765-4 functional OBDonCAN communication — Multiple-frame response not finished within $P2_{CAN}$

Figure 10 illustrates an example of a request message followed by one (1) multiple-frame response message and two (2) single frames (completion within $P2_{CAN\_max}$) and another request message. The next request message can be sent immediately by the external test equipment after completion of all response messages if the external test equipment knows the number of responding ECUs. If not, it needs to wait with the next request message to send until $P2_{CAN\_max}$ is expired.



Key

1 Client T_Data.req: diagnostic application issues functionally addressed request message to network layer.

2 All server T_Data.ind: network layer issues to diagnostic application the reception of a request message. All servers start the $P2_{CAN}$ timer using the value of $P2_{CAN} = P2_{CAN\_max}$.
Client T_Data.con: network layer issues to diagnostic application the confirmation of the completion of the request message. Client starts its $P2_{CAN}$ timer using the default reload value $P2_{CAN} = P2_{CAN\_max}$.

3 Server #1 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$.

4 Client T_DataSOM.ind: network layer issues to diagnostic application the reception of a StartOfMessage which is initiated by the reception of a FirstFrame indication on CAN (see ISO 15765-2). Client reloads $P2_{CAN}$ with $P2_{CAN\_max}$ value.

5 Server #2 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$. The response message is a single-frame message.

6 Server #2 T_Data.con: network layer issues to diagnostic application the completion of the response message.

Client T_Data.ind: network layer issues to diagnostic application the completion of the response message. Client reloads $P2_{CAN}$ with $P2_{CAN\_max}$ value.

7 Server #3 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$. The response message is a single-frame message.

8 Server #3 T_Data.con: network layer issues to diagnostic application the completion of the response message.

Client T_Data.ind: network layer issues to diagnostic application the completion of the response message. Client reloads $P2_{CAN}$ with $P2_{CAN\_max}$ value.

9 Server #1 T_Data.con: network layer issues to diagnostic application the completion of the response message.

Client T_Data.ind: network layer issues to diagnostic application the completion of the response message.

10 Client T_Data.req: diagnostic application issues functionally addressed request message to network layer. The client knows the number of ECUs which shall have responded to the previous request. Therefore, the client is not required to wait until the time window has reached $P2_{CAN\_max}$ before it issues a new functional request message.

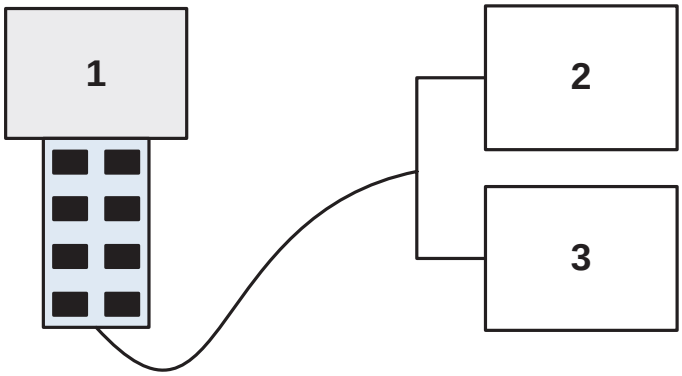
11 All server T_Data.ind: network layer issues to diagnostic application the reception of a request message. All servers start the $P2_{CAN}$ timer using the value of $P2_{CAN} = P2_{CAN\_max}$.
Client T_Data.con: network layer issues to diagnostic application the confirmation of the completion of the request message. Client starts its $P2_{CAN}$ timer using the default reload value $P2_{CAN} = P2_{CAN\_max}$.

Figure 10 — ISO 15765-4 functional OBDonCAN communication — Multiple-frame response finished within $P2_{CAN}$

NOTE The network layer timing parameters for the multiple-frame response are not shown. Network layer timing requirements for legislated diagnostic messages are specified in ISO 15765-4.

6.2.3.4 ECU behaviour to a request for supported/non-supported OBD information

Figure 11 illustrates an example of a typical vehicle OBD configuration.



Key

1 external test equipment
2 ECM (engine control module)
3 TCM (transmission control module)

Figure 11 — Example of external test equipment connected to two (2) OBD ECUs

A service shall only be implemented by an ECU if supported with data (e.g. PID/OBD Monitor ID/Test ID/INFOTYPE supported), except for Service 01_{16} and PID 00_{16} which shall be supported by all emissions-related ECUs.

Typically, the ECM supports OBD Monitor IDs, which the TCM does not support. In case the external test equipment requests the status of such OBD Monitor ID supported by the ECM, the ECM sends a positive response message and the TCM does not send a response message (no negative response message allowed). The external test equipment knows that the TCM will not send a positive response message based on the OBD Monitor ID supported information retrieved prior to the latter request.

This shall be implemented to enhance the overall diagnostic communication performance between the external test equipment and the vehicle ECUs (see 6.2.3.3).

6.2.4 Data not available

6.2.4.1 ISO 9141-2, ISO 14230-4, and SAE J1850 — Data not available

There are two conditions for which data are considered not available. One condition is that the service is not supported and the other is that the service is supported but data are currently not available.

For SAE J1850 and ISO 9141-2 interfaces, there will be no reject message to a functional request message if the request is not supported by the ECU. This prevents response messages from all ECUs that do not support a service or a specific data value.

For ISO 14230-4 interfaces, there will be a response message to every request message either positive (with data) or negative. In order to avoid unnecessary communication, the ECU(s) which does (do) not support a functionally requested PID, TID, or INFOTYPE is permitted not to send a negative response message because another ECU will send a positive response message. Format and possible codes of negative responses are specified in 6.3.4.

Some services are supported by a vehicle but data might not always be available when requested. For Services 05_{16} and 06_{16}, if the test has not been run since test results were cleared, or for Service 02_{16}, if freeze frame data has not been stored, or for Service 09_{16}, if the engine is running, valid data will not be available. For these conditions, the manufacturer has the option either to not respond or to respond with data that are invalid (ISO 9141-2 and SAE J1850 only). The functional description for these services discusses the method to determine if the data are valid.

6.2.4.2 ISO 15765-4 — Data not available

There are five (5) conditions for which data are considered not available:

a) Request message is not supported: The ECU(s) which does (do) not support the functional request message shall not send any response message;

b) Request message is supported but data are not supported: The ECU(s) which does (do) support the functional request message but does (do) not support the requested data (e.g. PID, OBD Monitor ID, TID, or INFOTYPE) is (are) not allowed to send a negative response message because another ECU will send a positive response message. If the external test equipment sends a message including multiple PIDs and each emission-related ECU does not support all requested PIDs, then each ECU shall send a positive response message including the supported PID(s) and data values and shall not send a negative response message. If an ECU does not support any of the PIDs requested, it is not allowed to send a negative response message;

c) Request message is supported but data are currently not available: The ECU(s) which does (do) support the functional request message but does (do) not currently have the requested data available shall respond with a negative response message with NRC 22_{16} - ConditionsNotCorrect (negative response message format is specified in 6.3.3). For Services 01_{16}, 02_{16}, 03_{16}, 06_{16}, 07_{16}, and $0A_{16}$, the use of a negative response message including NRC 22_{16} is not permitted. For Services 04_{16}, 08_{16}, and 09_{16}, the use of a NRC 22_{16} is allowed. Use of 22_{16} for Service 09_{16} CVN request may be restricted by OBD regulations;

d) Request message is supported but data are not available within P2 timing: The behaviour of the ECU(s) and the external test equipment is specified in 6.2.4.3;

e) Request message is supported but service cannot be performed within P2 timing: The behaviour of the ECU(s) and the external test equipment is specified in 6.2.4.3. For Services 04_{16} and 09_{16}, the use of an NRC 78_{16} is allowed.

6.2.4.3 Data not available within P2 timing

6.2.4.3.1 Overview

The following subclauses specify the request/response message handling for each protocol if the data are not available within the P2 timing in the ECU(s). The description in the subsection only applies to Service 09_{16}, INFOTYPE 06_{16} calibration verification numbers.

6.2.4.3.2 ISO 9141-2 — Data not available within P2 timing

If an ECU(s) supports the functional request message but does not have the requested data available within P2 timing, then a retry message handling routine shall be performed as follows:

a) If the response message is not received within $P2_{K\text{-}Line}$, the external test equipment shall stop retrying the request message after one (1) minute from the original request;

b) The retry message shall be sent at least every four (4) seconds (between 55 ms and 4 000 ms). The retry message keeps the bus alive and prevents the external test equipment from having to re-initialize the bus ($P3_{K\text{-}Line}$ time out);

c) The ECUs, which either have already sent a positive response message or have not sent a positive response message, shall not restart the requested internal routine again;

d) The external test equipment shall record if all ECUs have sent the expected number of response messages;

e) After successful completion of all response messages, the external test equipment shall send a request message which is "not equal" to the "Repeated Request" message.

Additional description is included in the functional description of the corresponding service.

Figure 12 illustrates the ISO 9141-2 (key bytes: 08_{16} 08_{16}) data not available within P2 timing handling overview.

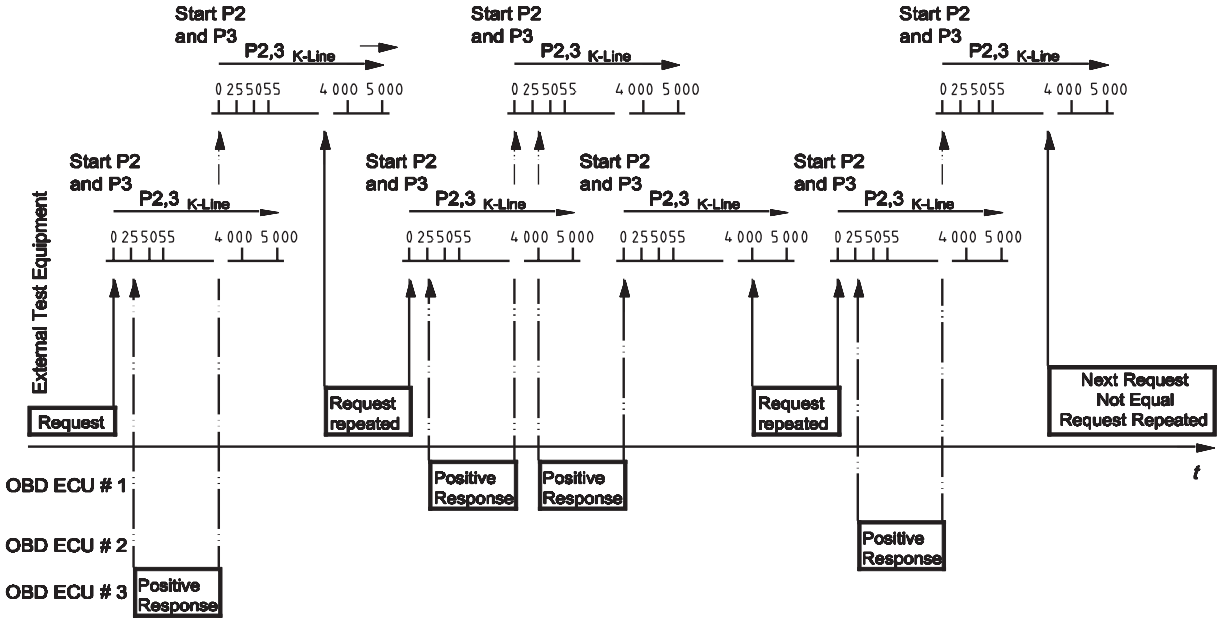


Figure 12 — ISO 9141-2 (key bytes: 08_{16} 08_{16}) data not available within P2 timing handling overview

For the ISO 9141-2 protocol, the response message timing $P2_{K\text{-}Line}$ shall be in accordance with Table 2. The $P2_{K\text{-}line\_min}$ application timing parameter value depends on the key bytes as listed:

— Key bytes: 08_{16} 08_{16}: $P2_{K\text{-}line\_min}$ = 25 ms;

— Key bytes: 94_{16} 94_{16}: $P2_{K\text{-}line\_min}$ = 0 ms.

6.2.4.3.3 ISO 14230-4 — Data not available within P2 timing

If an ECU(s) supports the functional request message but does not have the requested data available within P2 timing, handling shall be performed as follows:

a) The ECU(s) shall respond with a negative response message with NRC 78_{16} - RequestCorrectlyReceived-ResponsePending within P2 timing;

b) ECUs which require more time than $P2_{K\text{-}Line}$ to perform the requested action shall repeat the negative response message with NRC 78_{16} prior to expiration of P2K-Line until the positive response message is available;

c) After all positive response messages have been received or a time out $P2_{K\text{-}Line\_max}$ has occurred, the external test equipment shall wait until $P3_{K\text{-}Line\_min}$ is reached to send a new request message.

Figure 13 illustrates the ISO 14230-4 negative response code RC = 78_{16} handling overview.

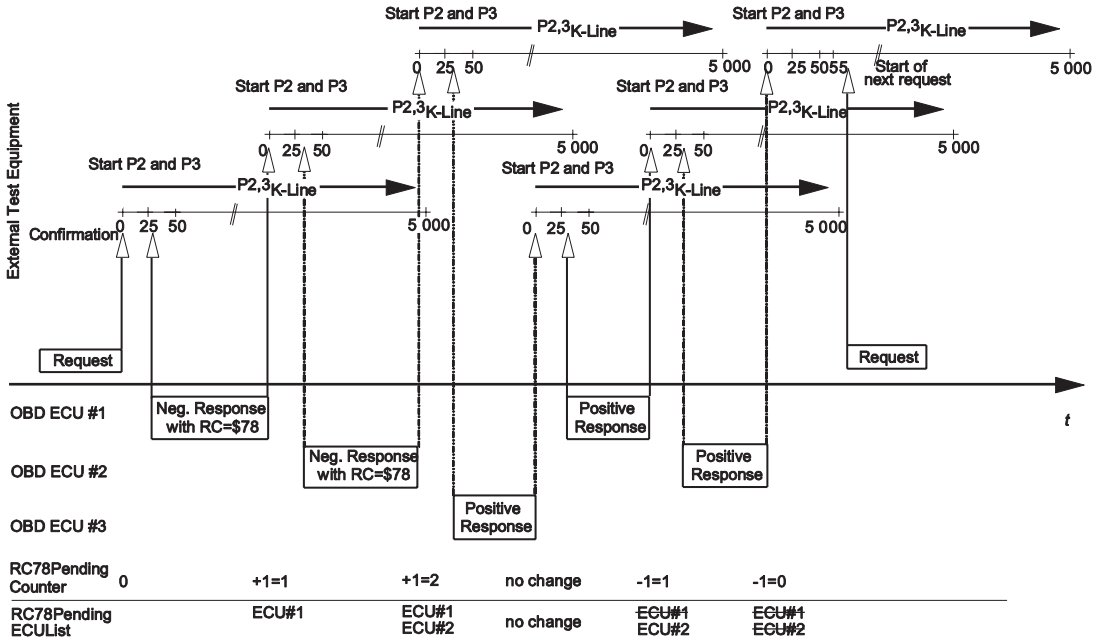


Figure 13 — ISO 14230-4:— Negative response code RC = 78_{16} handling overview

6.2.4.3.4 SAE J1850 — Data not available within P2 timing

If an ECU(s) supports the functional request message but does not have the requested data available within P2 timing, then a retry message handling routine shall be performed as follows:

a) If the response message is not received within $P2_{J1850}$, the external test equipment shall wait 30 (30 ± 1) s and then retry the request message;

b) The retry message shall be stopped after one (1) minute from the original request;

c) The external test equipment shall record if all ECUs have sent the expected number of response messages.

An additional description is included in the functional description of the corresponding service.

Figure 14 illustrates the SAE J1850 data not available within P2 timing handling overview.

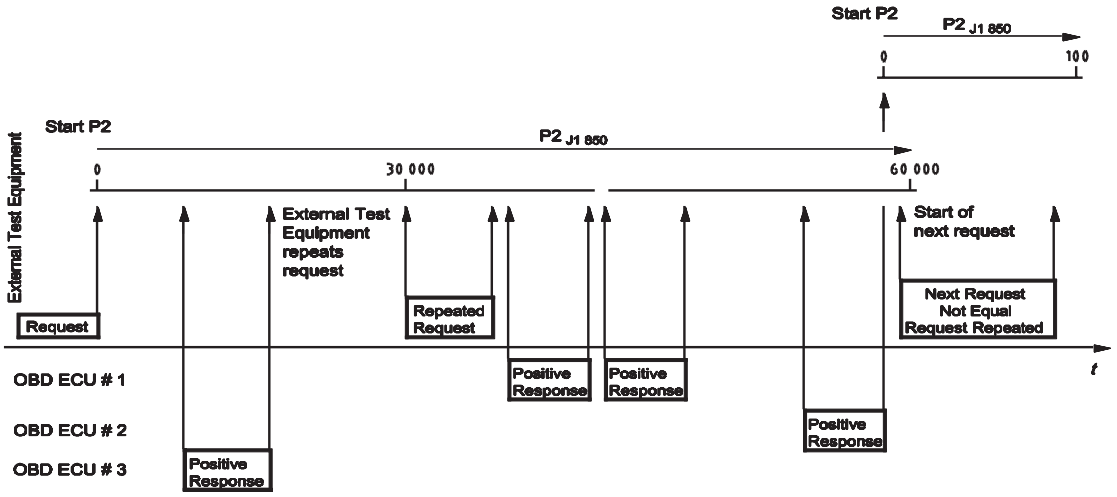


Figure 14 — SAE J1850 — Data not available within P2 timing handling overview

6.2.4.3.5 Data not available test conditions for protocols: ISO 9141-2, ISO 14230-4, and SAE J1850

There are two conditions for which data are considered not available:

a) Service is not supported;

b) Service is supported but data are not available at the time that the request is made.

Table 10 indicates the proper server/ECU response for each protocol as detailed in 6.2.4.1.

Table 10 — Proper response from server/ECU with ISO 9141-2, ISO 14230-4, and SAE J1850 protocol

Service	Condition	ISO 9141-2	SAE J1850	ISO 14230-4	NRC
00_{16} or $0A_{16}$-$0F_{16}$	Not allowed	The ECU shall not respond.	The ECU shall not respond.	ECU can either not respond or send a negative response.	11
01_{16}	Not supported	All ECUs shall respond to Service 01_{16} PID 00_{16} if Service 01_{16} is supported. If Service 01_{16} is not supported, no response is allowed.	All ECUs shall respond to Service 01_{16} PID 00_{16} if Service 01_{16} is supported. If Service 01_{16} is not supported, no response is allowed.	All ECUs shall respond to Service 01_{16} PID 00_{16} if Service 01_{16} is supported. If Service 01_{16} is not supported, ECU can either not respond or send a negative response.	11_{16}
	Unsupported PID requested	No response preferred, positive response is allowed.	No response preferred, positive response is allowed.	ECU can either not respond or send a negative response.	12_{16}
	Supported PID requested	Positive response is required.	Positive response is required.	Positive response is required.	N/A

Table 10 *(continued)*

Service	Condition	ISO 9141-2	SAE J1850	ISO 14230-4	NRC
02_{16}	Not supported	The ECU shall not respond.	The ECU shall not respond.	ECU can either not respond or send a negative response.	11_{16}
	Supported PID requested, no Freeze Frame stored	PID 02_{16} indicates 0000_{16}, but if PIDs are requested, ECU can either not respond or send invalid data, except if supported PIDs (00_{16}, 20_{16}, ...) have been requested, then the ECU shall send a response with the supported PID and data bytes.	PID 02_{16} indicates 0000_{16}, but if PIDs are requested, ECU can either not respond or send invalid data, except if supported PIDs (00_{16}, 20_{16}, ...) have been requested, then the ECU shall send a response with the supported PID and data bytes.	PID 02_{16} indicates 0000_{16}, but if PIDs are requested, ECU can either not respond or send a negative response, except if supported PIDs (00_{16}, 20_{16}, ...) have been requested, then the ECU shall send a response with the supported PID and data bytes.	12_{16}
	Unsupported PID requested, no Freeze Frame stored	No response preferred, positive response is allowed.	No response preferred, positive response is allowed.	ECU can either not respond or send a negative response.	12_{16}
	Supported PID requested, Freeze Frame stored	Positive response is required.	Positive response is required.	Positive response is required.	N/A
	Unsupported PID requested, Freeze Frame stored	No response preferred, positive response is allowed.	No response preferred, positive response is allowed.	ECU can either not respond or send a negative response.	12_{16}
03_{16} /07_{16}	Not supported	The ECU shall not respond.	The ECU shall not respond.	ECU can either not respond or send a negative response.	11_{16}
	Supported, no DTCs stored	No response preferred, positive response indicating no DTCs is allowed.	No response preferred, positive response indicating no DTCs is allowed.	Positive response indicating no DTCs is required.	N/A
	Supported, DTCs stored	Positive response is required.	Positive response is required.	Positive response is required.	N/A
04_{16}	Not supported	The ECU shall not respond.	The ECU shall not respond.	ECU can either not respond or send a negative response.	11_{16}
	Supported, conditions not correct	The ECU shall not respond.	The ECU shall not respond.	Negative response is required.	22_{16}
	Supported, conditions correct	Positive response is required.	Positive response is required.	Positive response is required.	N/A

Table 10 *(continued)*

Service	Condition	ISO 9141-2	SAE J1850	ISO 14230-4	NRC
05_{16} /06_{16}	Not supported	The ECU shall not respond.	The ECU shall not respond.	ECU can either not respond or send a negative response.	$_{16}11$
05_{16} /06_{16}	Supported TID requested, no stored data available	If TIDs are requested, ECU can either not respond or send invalid data.	If TIDs are requested, ECU can either not respond or send invalid data.	If TIDs are requested, ECU can either not respond or send invalid data or send negative response.	12_{16}
	Unsupported TID requested, no stored data available	No response preferred, positive response is allowed.	No response preferred, positive response is allowed.	ECU can either not respond or send a negative response.	12_{16}
	Supported TID requested, stored data available	Positive response is required.	Positive response is required.	Positive response is required.	N/A
	Unsupported TID requested, stored data available	No response preferred, positive response is allowed.	No response preferred, positive response is allowed.	ECU can either not respond or send a negative response.	12_{16}
08_{16}	Not supported	The ECU shall not respond.	The ECU shall not respond.	ECU can either not respond or send a negative response.	11_{16}
	Supported TID requested, conditions correct	Respond within P2 timing.	Respond within P2 timing.	Respond within P2 timing.	N/A
	Supported TID requested, conditions not correct	The ECU shall not respond or may respond with a manufacturer-specified value as DATA A, which corresponds to the reason the test cannot be run.	The ECU shall not respond or may respond with a manufacturer-specified value as DATA A, which corresponds to the reason the test cannot be run.	Negative response is required or may respond with a manufacturer-specified value as DATA A which corresponds to the reason the test cannot be run.	22_{16}
	Unsupported TID requested	No response preferred, positive response is allowed.	No response preferred, positive response is allowed.	ECU can either not respond or send a negative response.	12_{16}
09_{16}	Not supported	The ECU shall not respond.	The ECU shall not respond.	ECU can either not respond or send a negative response.	11_{16}
	Supported INFOTYPE requested, data available (VIN, CVN, CALID)	Positive response is required.	Positive response is required.	Positive response is required.	N/A

Table 10 *(continued)*

Service	Condition	ISO 9141-2	SAE J1850	ISO 14230-4	NRC
	Supported INFOTYPE requested, data not available, conditions correct (CVN)	Respond within 1 min; do not restart CVN calculation. Test tool sends retry message every 0,055 to 4,0 s.	Respond within 1 min; do not restart CVN calculation. Test tool sends retry message after 30 s.	One or multiple negative response message(s) required within $P2_{max}$ (25 – 50 ms) until positive response is sent.	78_{16}
	Supported INFOTYPE requested, data not available, conditions not correct (CVN), prior to 2005 MY only	The ECU shall not respond.	The ECU shall not respond.	Negative response is required.	22_{16}
	Unsupported INFOTYPE requested	No response preferred, positive response is allowed.	No response preferred, positive response is allowed.	ECU can either not respond or send a negative response.	12_{16}

NOTE 1 ISO 9141-2 and SAE J1850 do not support negative response codes.

NOTE 2 Negative response structure follows the scheme: $7F_{16}$, ServiceID, NRC (e.g. $7F_{16}$, 01_{16}, 11_{16}).

6.2.4.3.6 ISO 15765-4 — Data not available within P2 timing or service cannot be performed within P2 timing

The ECU(s) which does (do) support the functional request message but does (do) not have the requested data available within P2 timing or cannot perform the requested service within P2 timing shall perform the following handling:

a) The ECU(s) shall respond with a negative response message with NRC 78_{16} - RequestCorrectlyReceived-ResponsePending within P2 timing (not allowed for Service 01_{16}, 02_{16}, 03_{16}, 06_{16}, 07_{16}, and $0A_{16}$ requests);

b) After correct reception of the negative response message with NRC 78_{16}, the $P2_{CAN\_max}$ parameter timing value shall be set to $P2^*_{CAN\_max}$ (5 000 ms) by the external test equipment and the ECU which has sent the negative response message;

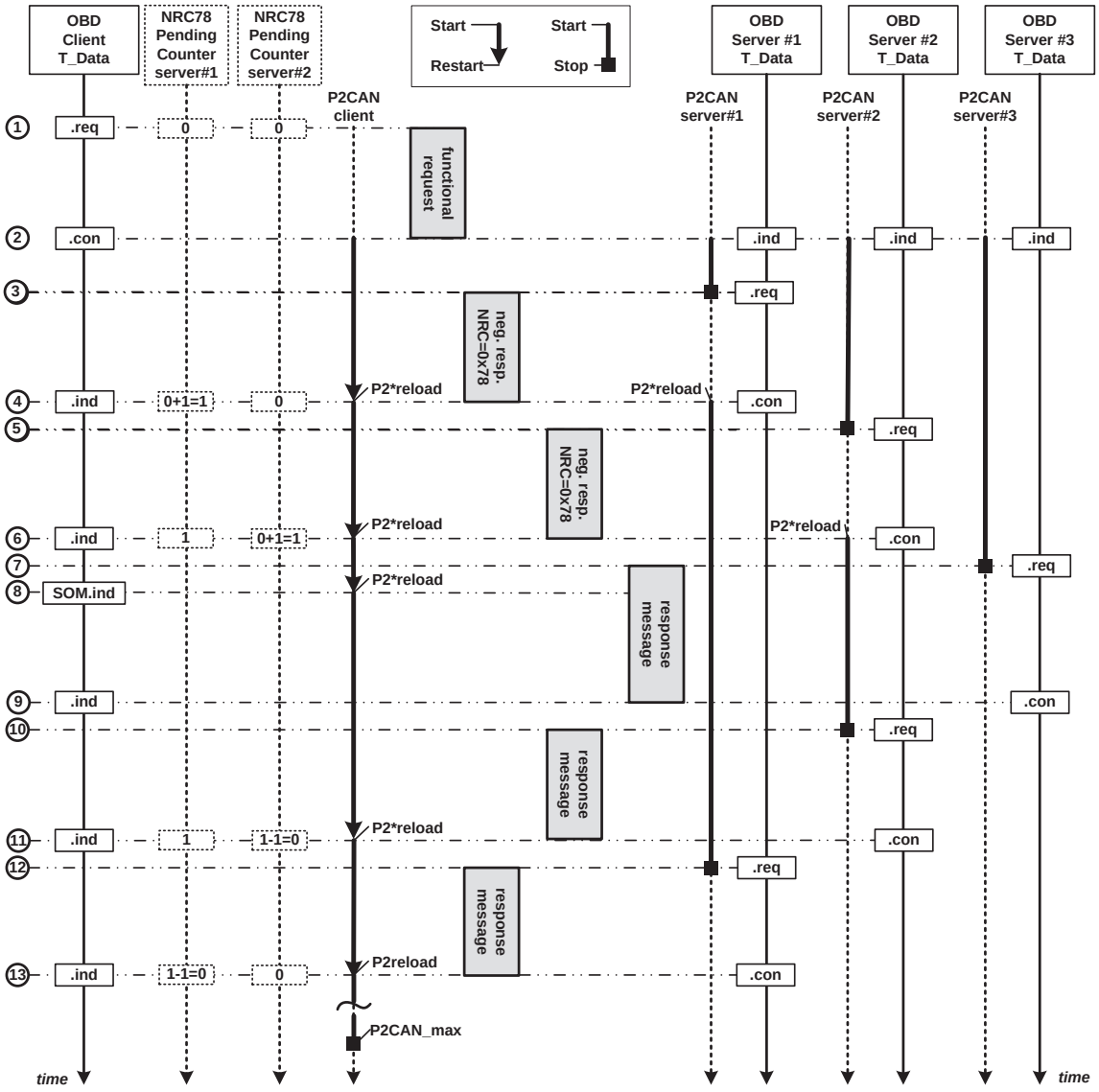
c) If another ECU also sends a negative response message with NRC 78_{16}, the $P2_{CAN\_max}$ timing parameter value shall be reloaded to $P2^*_{CAN\_max}$;

d) ECUs which require more than $P2^*_{CAN\_max}$ to perform the requested action shall repeat the negative response message with NRC 78_{16} prior to expiration of $P2^*_{CAN\_max}$ until correct reception of the positive response message;

e) After all positive response messages have been received or timed out, $P2^*_{CAN\_max}$ has occurred, the $P2_{CAN\_max}$ timing parameter shall be reset to the values specified in Table 7.

The vehicle manufacturer is responsible to ensure that the network architecture of the vehicle does not cause timing delays that exceed $P2_{CAN\_max}$ timing when responding to Services 01_{16}, 02_{16}, 03_{16}, 06_{16}, 07_{16}, 08_{16}, and $0A_{16}$ requests because a negative response message with NRC 78_{16} shall not be allowed.

Figure 15 illustrates the negative response message handling with NRC 78_{16} for the ISO 15765-4 interface.



Key

1 Client T_Data.req: diagnostic application issues functionally addressed request message to network layer.

2 All server T_Data.ind: network layer issues to diagnostic application the reception of a request message. All servers start the $P2_{CAN}$ timer using the value of $P2_{CAN} = P2_{CAN\_max}$.
Client T_Data.con: network layer issues to diagnostic application the confirmation of the completion of the request message. All NRCPendingCounter = 0. Client starts its $P2_{CAN}$ timer using the default reload value $P2_{CAN} = P2_{CAN\_max}$.

3 Server #1 T_Data.req: diagnostic application does not have the positive response message ready and issues negative response message with NRC = 78_{16} by a T_Data.req to the network layer within $P2_{CAN}$.

4 Client T_Data.ind: network layer issues to diagnostic application the reception of a message. Since the received response message is a negative response message with NRC = 78_{16}, the NRCPendingCounter for server #1 is incremented by 1 (0+1 = 1).
Client reloads $P2_{CAN}$ with $P2^*_{CAN\_max}$ value.
Server #1 T_Data.con: network layer issues to diagnostic application the completion of the response message.

5 Server #2 T_Data.req: diagnostic application does not have the positive response message ready and issues negative response message with NRC = 78_{16} by a T_Data.req to the network layer within $P2_{CAN}$

6 Client T_Data.ind: network layer issues to diagnostic application the reception of a message. Since the received response message is a negative response message with NRC = 78_{16}, the NRCPendingCounter server #2 is incremented by 1 (0+1 = 1).
Client reloads $P2_{CAN}$ with $P2^*_{CAN\_max}$ value.
Server #2 T_Data.con: network layer issues to diagnostic application the completion of the response message.

7 Server #3 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$.

8 Client T_DataSOM.ind: network layer issues to diagnostic application the reception of a StartOfMessage which is initiated by the reception of a FirstFrame indication on CAN (see ISO 15765-2). Client reloads $P2_{CAN}$ with $P2^*_{CAN\_max}$ value.

9 Server #3 T_Data.con: network layer issues to diagnostic application the completion of the response message.

Client T_Data.ind: network layer issues to diagnostic application the completion of the response message.

10 Server #2 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$. The response message is a single-frame message.

11 Server #2 T_Data.con: network layer issues to diagnostic application the completion of the response message.

Client T_Data.ind: network layer issues to diagnostic application the completion of the response message. Since the received response message is a positive response message, the NRCPendingCounter server #2 is decremented by 1 (1-1 = 0). Client reloads $P2_{CAN}$ with $P2^*_{CAN\_max}$ value.

12 Server #1 T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{CAN}$. The response message is a single-frame message.

13 Server #1 T_Data.con: network layer issues to diagnostic application the completion of the response message.

Client T_Data.ind: network layer issues to diagnostic application the completion of the response message. Since the received response message is a positive response message, the NRCPendingCounter server #1 is decremented by 1 (1-1 = 0). Client reloads $P2_{CAN}$ with $P2_{CAN\_max}$ value.

Figure 15 — ISO 15765-4:— Negative response code NRC = 78_{16} handling overview

6.2.4.3.7 Data not available — Test conditions for protocol: ISO 15765-4 Diagnostic communication over CAN

The following are the five (5) conditions for which data are considered not available:

— service is not supported;

— service is supported but data are not supported;

— service is supported but data are not available at the time that the request is made;

— service is supported but data are not available within P2 timing;

— service is supported but cannot be performed within P2 timing.

Table 11 indicates the proper server/ECU response as detailed in 6.2.4.2.

Table 11 — Proper response from server/ECU for ISO 15765-4 protocol

Service	Condition	ISO 15765-4	NRC
01_{16}	Not supported	All ECUs shall respond to Service 01_{16} PID 00_{16} if Service 01_{16} is supported. If Service 01_{16} is not supported, no response is allowed.	N/A
	Unsupported PID requested	The ECU shall not respond.	N/A
	Supported PID requested	Positive response is required (no negative response message with NRC 78_{16} allowed).	N/A
	Support PID 00_{16} requested during initialization	Positive response is required or negative response for max of 5 times.	21_{16}

Table 11 *(continued)*

Service	Condition	ISO 15765-4	NRC
02_{16}	Not supported	The ECU shall not respond.	N/A
	Supported PID, frame XX_{16} requested, no Freeze Frame stored	1) The ECU shall respond to PID 02_{16} frame XX_{16}; PID 02_{16} frame XX_{16} shall indicate 0000_{16}. 2) The ECU shall respond with support PIDs for frame XX_{16} (00_{16}, 20_{16}, ...). 3) If PIDs other than support PIDs or PID 02_{16} are requested, the ECU shall not respond.	N/A
	Unsupported PID, frame XX_{16} requested, no Freeze Frame stored	PID 02_{16} frame XX_{16} indicates 0000_{16}, but if any other PIDs are requested, ECU shall not respond.	N/A
	Supported PID, frame XX_{16} requested, Freeze Frame stored	1) The ECU shall respond to PID 02_{16} frame XX_{16} within P2 timing. 2) The ECU shall respond with support PIDs for frame XX_{16} (00_{16}, 20_{16} ...) within P2 timing and shall respond to PIDs frame XX_{16} indicated as supported within P2 timing.	N/A
	Unsupported PID, frame XX_{16} requested, Freeze Frame stored	The ECU shall not respond.	N/A
03_{16} / 07_{16} / $0A_{16}$	Not supported	The ECU shall not respond.	N/A
	Supported, no DTCs stored	Positive response indicating no DTCs is required.	N/A
	Supported, DTCs stored	Positive response including the stored DTCs is required.	N/A
04_{16}	Not supported	The ECU shall not respond.	N/A
	Supported, conditions not correct	Negative response is required.	22_{16}
	Supported, conditions correct	Positive response message required. Multiple negative response messages allowed within a maximum time of 5 000 ms after request until positive response is required.	78
06_{16}	Not supported	The ECU shall not respond.	N/A
	Supported OBDMID requested, no stored data available	Positive response required; test values, min and max limits shall be set to 00_{16}.	N/A
	Unsupported OBDMID requested, no stored data available	The ECU shall not respond.	N/A
	Supported OBDMID requested, stored data available	Positive response is required.	N/A
	Unsupported OBDMID requested, stored data available	The ECU shall not respond.	N/A
08_{16}	Not supported	The ECU shall not respond.	N/A
	Supported TID requested, conditions correct	Positive response is required.	N/A
	Supported TID requested, conditions not correct	Negative response required.	22_{16}
	Unsupported TID requested	The ECU shall not respond.	N/A

Table 11 *(continued)*

Service	Condition	ISO 15765-4	NRC
09_{16}	Not supported	The ECU shall not respond.	N/A
	Supported INFOTYPE requested, data available (VIN, CVN, CALID)	Positive response is required.	N/A
	Supported INFOTYPE requested, data not available, conditions correct (CVN)	Initial negative response message required within $P2_{max}$ (50 ms) and consecutive negative response message(s) is (are) required within $P2_{max}$ (5.0 s) until positive response is sent.	78_{16}
	Supported INFOTYPE requested, data not available, conditions not correct (CVN), prior to 2005 MY only	Negative response is required. (Use of NRC 22_{16} may be restricted by OBD regulations.)	22_{16}
	Unsupported INFOTYPE requested	The ECU shall not respond.	N/A
00_{16}, 05_{16} or $0B_{16}$ – $0F_{16}$	Not allowed	The ECU shall not respond.	N/A

NOTE Negative response structure follows the scheme: $7F_{16}$, ServiceID, NRC (e.g. $7F_{16}$, 01_{16}, 11_{16}).

6.2.5 Maximum values

If the data value exceeds the maximum value possible to be sent, the on-board system shall send the maximum value possible (FF_{16} or $FFFF_{16}$). The external test equipment shall display the maximum value or an indication of data too high. This is not normally critical for real-time diagnostics, but, for example, in the case of a misfire at high vehicle speed with resulting freeze frame data stored, this will be very valuable diagnostic information.

6.2.6 Invalid signals

In distributed network architectures, certain OBD devices may be hardwired to other ECUs or may be independent OBD mechatronic devices, e.g. smart sensor/actuator connected through a network from another ECU (both referred to as remote OBD devices). When remote OBD devices are not hardwired to the OBD ECU and the data are not received over the data bus from the specific remote OBD device, this might occur for two reasons; either the remote ECU is not functioning and sending any data, or the OBD device that is hardwired to the remote ECU has failed and the remote ECU is sending a message with invalid data for the OBD remote device. In either one of these cases, the primary OBD ECU shall report Service 01_{16} and Service 02_{16} data parameters as the minimum or maximum value to indicate that the signal has not been received. A PID which includes this invalid data (no signal) shall either be reported with a minimum value (00_{16} or 0000_{16}) or maximum value (FF_{16} or $FFFF_{16}$), e.g. PID $0D_{16}$ "Vehicle Speed Sensor" = FF_{16} = 255 km/h, PID $2F_{16}$ "Fuel Level Input" = 00_{16} = 0,0 %. The reported value shall be determined by the manufacturer based on system design and network architecture to represent the least likely value to be expected under normal conditions.

6.3 Diagnostic message format

6.3.1 Addressing method

Functional addressing shall be used for all request messages because the external test equipment does not know which system on the vehicle has the information that is needed.

6.3.2 Maximum message length

6.3.2.1 ISO 9141-2, ISO 14230-4, SAE J1850 — Maximum message length

The maximum message length for request and response messages is limited to seven (7) data bytes.

For SAE J1850 and ISO 9141-2 interfaces, each unique diagnostic message specified in this part of ISO 15031 is a fixed length, although not all messages are the same length. For Services 01_{16} and 02_{16}, message length is determined by parameter identification (PID). Several PIDs, e.g. 06_{16} – 09_{16}, require reading of PIDs 13_{16} and/or $1D_{16}$ to determine whether a data byte B is included in the response message. For Service 05_{16}, message length is determined by Test ID. For other services, the message length is determined by the service. This enables the external test equipment to check for proper message length and to recognize the end of the message without waiting for possible additional data bytes. For ISO 14230-4 interfaces, the message length is always determined by the length information included in the first byte of the header.

6.3.2.2 ISO 15765-4 — Maximum message length

The maximum message length is specified in ISO 15765-4. For request messages, the message length is limited to seven (7) data bytes.

6.3.3 Request/Response message format

6.3.3.1 ISO 9141-2, ISO 14230-4, SAE 1850, ISO 15765-4 — Request message format

Table 12 specifies the request message format.

Table 12 — Request message format for ISO 9141-2, ISO 14230-4, SAE J1850, ISO 15765-4

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request Service Identifier	M	XX_{16}	SIDRQ
#2	service-specific data byte#1	U	XX_{16}	—
#3	service-specific data byte#2	U	XX_{16}	—
#4	service-specific data byte#3	U	XX_{16}	—
#5	service-specific data byte#4	U	XX_{16}	—
#6	service-specific data byte#5	U	XX_{16}	—
#7	service-specific data byte#6	U	XX_{16}	—

The message format defined for some services for the ISO 15765-4 protocol allows for an optional number of data bytes in the request message sent by the external test equipment. If these are included in the request message, support of those optional data bytes becomes mandatory for the server/ECU.

6.3.3.2 ISO 9141-2, ISO 14230-4, SAE J1850 — Positive response message format

Table 13 specifies the positive response message format.

Table 13 — Positive response message format for ISO 9141-2, ISO 14230-4, SAE J1850

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Positive Response Service Identifier	M	XX_{16}	SIDPR
#2	service-specific data byte#1	U	XX_{16}	—
#3	service-specific data byte#2	U	XX_{16}	—
#4	service-specific data byte#3	U	XX_{16}	—
#5	service-specific data byte#4	U	XX_{16}	—
#6	service-specific data byte#5	U	XX_{16}	—
#7	service-specific data byte#6	U	XX_{16}	—

6.3.3.3 ISO 15765-4 — Positive response message format

Table 14 specifies the positive response message format.

Table 14 — Positive response message format for ISO 15765-4

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Positive Response Service Identifier	M	XX_{16}	SIDPR
#2	service-specific data byte#1	U	XX_{16}	—
#3	service-specific data byte#2	U	XX_{16}	—
#4	service-specific data byte#3	U	XX_{16}	—
:	:	:	:	:
#n-2	service-specific data byte#m-2	U	XX_{16}	—
#n-1	service-specific data byte#m-1	U	XX_{16}	—
#n	service-specific data byte#m	U	XX_{16}	—
n: This value depends on the response message length.				
m: This value depends on the response message length – 1.				

6.3.3.4 ISO 14230-4, ISO 15765-4 — Negative response message format

This subclause includes additions, exceptions, and/or restrictions for ISO 14230-4 and ISO 15765-4.

Table 15 specifies the negative response message format.

Table 15 — Negative response message format for ISO 14230-4, ISO 15765-4

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Negative Response Service Identifier	M	$7F_{16}$	SIDNR
#2	Request Service Identifier	M	XX_{16}	SIDRQ
#3	Response Code	M	XX_{16}	RC_

6.3.4 Response code parameter definition

Response codes shall be implemented in an ECU that supports a service(s) not having valid data available at the time of a request or which cannot respond with valid data available within $P2_{K\text{-}Line}$ and $P2_{CAN}$ timing.

Table 16 defines negative response codes.

Table 16 — Negative response code (NRC) definition

Supported by ISO Protocol	Byte Value	Definition of Response Code	Mnemonic
14230–4	10_{16}	generalReject This response code indicates that the service is rejected but the server (ECU) does not specify the reason of the rejection.	GR
14230–4	11_{16}	serviceNotSupported This response code indicates that the requested action will not be taken because the server (ECU) does not support the requested service.	SNS
14230–4	12_{16}	subFunctionNotSupported-InvalidFormat This response code indicates that the requested action will not be taken because the server (ECU) does not support the arguments of the request message or the format of the argument bytes does not match the prescribed format for the specified service.	SFNSIF

Table 16 *(continued)*

Supported by ISO Protocol	Byte Value	Definition of Response Code	Mnemonic
14230–4	21_{16}	busy-RepeatRequest	BRR
15765–4		This response code indicates that the server (ECU) is temporarily too busy to perform the requested operation. For ISO 15765-4 protocol, the client (external test equipment) shall behave as defined in ISO 15765-4. In a multi-client (more than one external test equipment, e.g. telematic client) environment, the diagnostic request message of one client might be blocked temporarily by a negative response message with response code 21_{16} while another client finishes a diagnostic task. Therefore, this negative response code (NRC) is only allowed to be used during the initialization sequence of the protocol. NOTE If the server (ECU) is able to perform the diagnostic task but needs additional time to finish the task and prepares the response message, the negative response message with response code 78_{16} is used instead of 21_{16}.	
14230–4 15765–4	22_{16}	conditionsNotCorrectOrRequestSequenceError This response code indicates that the requested action will not be taken because the server (ECU) prerequisite conditions are not met. This request might also occur when sequence-sensitive requests are issued in the wrong order.	CNCORSE
14230–4 15765–4	78_{16}	requestCorrectlyReceived-ResponsePending This response code indicates that the request message was received correctly and that any parameters in the request message were valid but the action to be performed may not be completed yet. This response code can be used to indicate that the request message was properly received and does not need to be re-transmitted but the server (ECU) is not yet ready to receive another request. The negative response message with this response code may be repeated by the ECU(s) within P2K-Line = $P2_{CAN}$ = $P2^*_{max}$ until the positive response message with the requested data is available.	RCR-RP

6.3.5 Header byte definition of ISO 9141-2, ISO 14230-4, and SAE J1850

The first three (3) bytes of all diagnostic messages are the header bytes.

For SAE J1850 and ISO 9141-2 interfaces, the value of the first header byte is dependent on the bit rate of the data link and the type of message (see SAE J1850 and ISO 9141-2). The second header byte has a value that depends on the type of message, either a request or a response.

For ISO 14230-4 interfaces, the value of the first header byte indicates the addressing mode (physical/functional) and the length of the data field. The second header byte is the address of the receiver of the message. The third header byte for all interfaces is the physical address of the sender of the message. The external test equipment has the address $F1_{16}$. Other service tools shall use addresses in the range from $F0_{16}$ to FD_{16}. The response to all request messages will be independent of the address of the external test equipment requesting the information. Vehicle manufacturers shall not use the header bytes defined in this part of ISO 15031 for any purpose other than emissions-related diagnostic messages. When they are used, they shall conform to this specification.

Table 17 defines the diagnostic message format for ISO 9141-2, ISO 14230-4, and SAE J1850 protocols.

Table 17 — Diagnostic message format for ISO 9141-2, ISO 14230-4, SAE J1850

Header Bytes (Hex)			Data Bytes								
Priority/Type	Target Address (hex)	Source Address (hex)	#1	#2	#3	#4	#5	#6	#7	ERR	RESP
Diagnostic Request at 10,4 kbit/s: SAE J1850 and ISO 9141-2											
68_{16}	$6A_{16}$	$F1_{16}$	Maximum 7 data bytes							Yes	No
Diagnostic Response at 10,4 kbit/s: SAE J1850 and ISO 9141-2											
48_{16}	$6B_{16}$	ECU addr	Maximum 7 data bytes							Yes	No
Diagnostic Request at 10,4 kbit/s (ISO 14230-4)											
$11LL\ LLLL_2$	33_{16}	$F1_{16}$	Maximum 7 data bytes							Yes	No
Diagnostic Response at 10,4 kbit/s (ISO 14230-4)											
$10LL\ LLLL_2$	$F1_{16}$	ECU addr	Maximum 7 data bytes							Yes	No
Diagnostic Request at 41,6 kbit/s (SAE J1850)											
61_{16}	$6A_{16}$	$F1_{16}$	Maximum 7 data bytes							Yes	Yes
Diagnostic Response at 41,6 kbit/s (SAE J1850)											
41_{16}	$6B_{16}$	ECU addr	Maximum 7 data bytes							Yes	Yes

NOTE LL LLLL = Length of data bytes; RESP = In-frame response; ERR = Error detection.

6.3.6 Header byte definition of ISO 15765-4

Each CAN frame is identified by a CAN Identifier. The size of the identifier is either 11 bit or 29 bit. The CAN identifier shall always be followed by an eight (8) byte CAN frame data field [see ISO 15765-4; section “Data length code (DLC)”]. Depending on the message type, up to three (3) bytes (FlowControl) are used for the PCI (protocol control information) prior to the service identifier (only included in single frame or first frame) and data bytes of the message.

Table 18 defines the diagnostic message format for ISO 15765-4 protocol.

Table 18 — Diagnostic message format for ISO 15765-4

Header Bytes	CAN frame data field							
CAN Identifier (11 or 29 bit)	#1	#2	#3	#4	#5	#6	#7	#8

6.3.7 Data bytes definition of ISO 9141-2, ISO 14230-4, SAE J1850, and ISO 15765-4

For the ISO 9141-2, ISO 14230-4, and the SAE J1850 protocol, the first data byte following the header is the diagnostic service identifier and the remaining data bytes vary depending on the specific diagnostic service. For the ISO 15765-4 protocol, the first data byte following the CAN Identifier in a single frame and first frame is the PCI (protocol control information, number of bytes varies, depending on frame type), then diagnostic service identifier, and the remaining data bytes vary depending on the specific diagnostic service.

6.3.8 Non-data bytes included in diagnostic messages with SAE J1850

All diagnostic messages use a cyclic redundancy check (CRC) as in SAE J1850 as the error detection byte (ERR). In-frame response (RSP) is specified as optional in SAE J1850. For messages specified in this part of ISO 15031, the RSP byte is required in all request and response messages at 41,6 kbit/s and is not allowed for messages at 10,4 kbit/s. The in-frame response byte shall be the node address of the device transmitting the RSP. SAE J1850 specifies additional message elements that may be included in diagnostic messages. Use of these message elements is beyond the scope of this part of ISO 15031 but needs to be considered when specifying total diagnostic messages.

6.3.9 Non-data bytes included in diagnostic messages with ISO 9141-2 and ISO 14230-4

Messages will include a checksum, specified in ISO 9141-2 and ISO 14230-4, after the data bytes as the error detection byte (ERR). There is no provision for an in-frame response.

In the bit position convention, some data byte values include descriptions that are based on bit positions within the byte. The convention used is that the most significant bit (MSB) is referred to as "bit 7" and the least significant bit (LSB) is referred to as "bit 0," as shown in Figure 16.

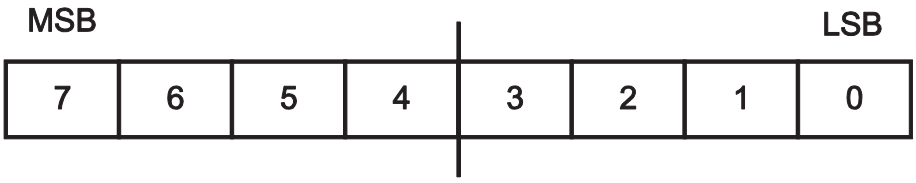


Figure 16 — Bit position within a data byte

6.4 Byte order convention

When reporting data larger than one byte, the Most Significant Byte (or high byte) is reported as first data byte followed by the next most significant bytes. The Least Significant Byte (or low byte) is reported as the last data byte. This convention is shown in numerous examples throughout this part of ISO 15031.

6.5 Allowance for expansion and enhanced diagnostic services

This part of ISO 15031 allows for the addition of diagnostic services both as industry standards and manufacturer-specific services. The diagnostic services 00_{16} through $0F_{16}$ are ISO/SAE reserved.

6.6 Definition of PIDs for services 01_{16} and 02_{16}

All PIDs are defined in SAE J1979-DA.

IMPORTANT — Several new PIDs have been defined in SAE J1979-DA based on new emissions-related vehicle technology. The data size of those PIDs exceeds the maximum message length of the non-CAN protocols (SAE J1850, ISO 9141-2, ISO 14230-4). Those PIDs are not supported by the non-CAN protocols.

6.7 Format of data to be displayed

Table 19 indicates the type of data and minimum requirements for the display format.

Table 19 — Format of data to be displayed

Data	Services	Display Format
Device ID – source address of response	All	ISO 9141-2: Hexadecimal (00_{16} to FF_{16}) ISO 14230-4: Hexadecimal (00_{16} to FF_{16}) SAE J1850: Hexadecimal (00_{16} to FF_{16}) ISO 15765-4: Hexadecimal (11 bit or 29 bit CAN Identifier)
Parameter ID (PID)	01_{16} and 02_{16}	Hexadecimal (00_{16} to FF_{16}) description (see SAE J1979-DA)
Frame number	02_{16}	Decimal (0 to 255)
Data values	01_{16} and 02_{16}	See SAE J1979-DA
Diagnostic trouble codes	03_{16}, 07_{16}, and $0A_{16}$	"P", "B", "C", or "U", plus 4 hexadecimal characters and/or DTC definition (see SAE J2012-DA)

Table 19 *(continued)*

Data	Services	Display Format
Test ID	05_{16}, 06_{16}, and 08_{16}	Hexadecimal (00_{16} to FF_{16})
Test value and test limits	05_{16}	Engineering units for Test IDs less than 80_{16} (see SAE J1979-DA) – decimal (0 to 255) for Test IDs greater than 80_{16}
Test value and test limits	06_{16}	Decimal (0 to 65 535)
Component ID	06_{16}	Hexadecimal (00_{16} to $7F_{16}$)
Optional data bytes	08_{16}	4 bytes, each decimal (0 to 255) (see SAE J1979-DA)
Vehicle information type	09_{16}	Hexadecimal (00_{16} to $7F_{16}$) (see SAE J1979-DA)
Vehicle information data	09_{16}	See SAE J1979-DA

NOTE ISO 15031-4/SAE J1978 specifies further guidelines and examples on displaying Service 01_{16} through 09_{16} data.

7 Diagnostic service definition for ISO 9141-2, ISO 14230-4, and SAE J1850

7.1 Service 01_{16} — Request current powertrain diagnostic data

7.1.1 Functional description

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 parameter identification (PID) value that indicates to the on-board system the specific information requested. PID specifications, scaling information, and display formats are included in SAE J1979-DA.

The ECU(s) shall respond to this message by transmitting the requested data value last determined by the system. All data values returned for sensor readings will be actual readings, not default or substitute values used by the system because of a fault with that sensor.

Not all PIDs are applicable or supported by all systems. PID 00_{16} is a bit-encoded PID that indicates, for each ECU, which PIDs that ECU supports. PID 00_{16} shall be supported by all ECUs that respond to a Service 01_{16} request because the external test equipment that conforms to SAE J1978 uses the presence of a response message by the vehicle to this request message to determine which protocol is supported for diagnostic communications. SAE J1979-DA defines how to encode supported PIDs.

IMPORTANT — All emissions-related OBD ECUs which support at least one of the services defined in this part of ISO 15031 shall support Service 01_{16} and PID 00_{16}. Service 01_{16} with PID 00_{16} is defined as the universal "initialization/keep alive/ping" message for all emissions-related OBD ECUs.

7.1.2 Message data bytes

7.1.2.1 Request current powertrain diagnostic data request message definition (read-supported PIDs)

Table 20 — Request current powertrain diagnostic data request message (read-supported PIDs)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request current powertrain diagnostic data request SID	M	01_{16}	SIDRQ
#2	PID (see SAE J1979-DA)	M	XX_{16}	PID

7.1.2.2 Request current powertrain diagnostic data response message definition (report supported PIDs)

Table 21 — Request current powertrain diagnostic data response message (report supported PIDs)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request current powertrain diagnostic data response SID	M	41_{16}	SIDPR
	data record of supported PID = [			PIDREC_
#2	supported PID	M	XX_{16}	PID
#3	data A,	M	XX_{16}	DATA_A
#4	data B,	M	XX_{16}	DATA_B
#5	data C,	M	XX_{16}	DATA_C
#6	data D]	M	XX_{16}	DATA_D

7.1.2.3 Request current powertrain diagnostic data request message definition (read PID value)

Table 22 — Request current powertrain diagnostic data request message (read PID value)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request current powertrain diagnostic data request SID	M	01	SIDRQ
#2	PID (see SAE J1979-DA)	M/C[a]	XX_{16}	PID
[a] C = Conditional — PID value is one of the supported PIDs of previous response message.				

7.1.2.4 Request current powertrain diagnostic data response message definition (report PID value)

Table 23 — Request current powertrain diagnostic data response message (report PID value)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request current powertrain diagnostic data response SID	M	41_{16}	SIDPR
	data record of 1st supported PID = [			PIDREC_
#2	PID	M	XX_{16}	PID
#3	data A,	M	XX_{16}	DATA_A
#4	data B,	C[a]	XX_{16}	DATA_B
#5	data C,	C	XX_{16}	DATA_C
#6	data D]	C	XX_{16}	DATA_D
[a] C = Conditional — data B - D depend on selected PID value.				

The PID, which is included in the request message, may be supported by all emission-related ECUs, which shall comply with this specification. Therefore, multiple response messages are sent by the vehicle ECUs.

7.1.3 Parameter definition

7.1.3.1 PIDs supported

SAE J1979-DA specifies the interpretation of the data record of supported PIDs.

7.1.3.2 PID and data byte descriptions

SAE J1979-DA specifies standardized emission-related parameters.

7.1.4 Message example

The example below shows how the "Request current powertrain diagnostic data" service shall be implemented.

7.1.4.1 Step #1: Request supported PIDs from vehicle

The external test equipment requests supported PIDs (PID = 00_{16}, 20_{16}) from the vehicle. Refer to SAE J1979-DA to interpret the data bytes in the response messages.

Table 24 — Request current powertrain diagnostic 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 current powertrain diagnostic data request SID	01_{16}	SIDRQ
#2	PID used to determine PID support for PIDs 01_{16} - 20_{16}	00_{16}	PID

Table 25 — Request current powertrain diagnostic 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 current powertrain diagnostic data response SID	41_{16}	SIDPR
#2	PID requested	00_{16}	PID
#3	Data byte A, representing support for PIDs 01_{16}, 03_{16} – 08_{16}	$10111111_2 = BF_{16}$	DATA_A
#4	Data byte B, representing support for PIDs 09_{16}, $0B_{16}$ – 10_{16}	$10111111_2 = BF_{16}$	DATA_B
#5	Data byte C, representing support for PIDs 11_{16}, 13_{16}, 15_{16}	$10101000_2 = A8_{16}$	DATA_C
#6	Data byte D, representing support for PIDs 19_{16}, $1C_{16}$, 20_{16}	$10010001_2 = 91_{16}$	DATA_D

Table 26 — Request current powertrain diagnostic data response message

Message Direction:	ECU#2 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request current powertrain diagnostic data response SID	41_{16}	SIDPR
#2	PID requested	00_{16}	PID
#3	Data byte A, representing support for PID 01_{16}	10000000b = 80_{16}	DATA_A
#4	Data byte B, representing support for PID $0D_{16}$	00001000b = 08_{16}	DATA_B
#5	Data byte C, representing no support for PIDs 11_{16} – 18_{16}	00000000b = 00_{16}	DATA_C
#6	Data byte D, representing no support for PIDs 19_{16} – 20_{16}	00000000b = 00_{16}	DATA_D

Table 27 — Request current powertrain diagnostic 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 current powertrain diagnostic data request SID	01_{16}	SIDRQ
#2	PID requested	20_{16}	PID

Table 28 — Request current powertrain diagnostic 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 current powertrain diagnostic data response SID	41_{16}	SIDPR
#2	PID requested	20_{16}	PID
#3	Data byte A, representing support for PID 21_{16}	10000000b = 80_{16}	DATA_A
#4	Data byte B, representing no support for PIDs 29_{16} – 30_{16}	00000000b = 00_{16}	DATA_B
#5	Data byte C, representing no support for PIDs 31_{16} – 38_{16}	00000000b = 00_{16}	DATA_C
#6	Data byte D, representing no support for PIDs 39_{16} – 40_{16}	00000000b = 00_{16}	DATA_D

NOTE ECU#2 does not send a response message because it indicated with the previous response message that it does not support PID 20_{16}.

Now the external test equipment creates an internal list of supported PIDs for each ECU. The ECU#1 (ECM) supports the following PIDs: 01_{16}, 03_{16} – 09_{16}, $0B_{16}$ – 11_{16}, 13_{16}, 15_{16}, 19_{16}, $1C_{16}$, 20_{16}, 21_{16}. The ECU#2 (TCM) supports the PIDs 01_{16} and $0D_{16}$.

7.1.4.2 Step #2: Request PID from vehicle

The external test equipment requests the following PID from the vehicle:

— PID 01_{16}: Number of emission-related powertrain DTCs and MIL status, PID is supported by ECU#1 (ECM) and ECU#2 (TCM)

Table 29 — Request current powertrain diagnostic 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 current powertrain diagnostic data request SID	01_{16}	SIDRQ
#2	PID: Number of emission-related powertrain DTCs and MIL status	01_{16}	PID

Table 30 — Request current powertrain diagnostic 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 current powertrain diagnostic data response SID	41_{16}	SIDPR
#2	PID: Number of emission-related powertrain DTCs and MIL status	01_{16}	PID
#3	MIL: ON; Number of emission-related powertrain DTCs: 01_{16}	81_{16}	DATA_A
#4	Misfire -, Fuel system -, Comprehensive monitoring	07_{16}	DATA_B

Table 30 *(continued)*

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#5	Catalyst -, Heated catalyst -, ..., monitoring supported	EF_{16}	DATA_C
#6	Catalyst -, Heated catalyst -, ..., monitoring test complete/not complete	63_{16}	DATA_D

Table 31 — Request current powertrain diagnostic data response message

Message Direction:	ECU#2 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request current powertrain diagnostic data response SID	41_{16}	SIDPR
#2	PID: Number of emission-related powertrain DTCs and MIL status	01_{16}	PID
#3	MIL: OFF; Number of emission-related powertrain DTCs: 01_{16}	01_{16}	DATA_A
#4	Comprehensive monitoring: supported, test complete	04_{16}	DATA_B
#5	Catalyst -, Heated catalyst -, ..., monitoring supported	00_{16}	DATA_C
#6	Catalyst -, Heated catalyst -, ..., monitoring test complete/not complete	00_{16}	DATA_D

The ECU#1 (ECM) reports MIL commanded on, one stored DTC, all monitors as supported, catalyst, heated catalyst, oxygen sensor and oxygen sensor heater as not completed, and all other monitors as completed.

The ECU#2 (TCM) reports MIL commanded off, one stored DTC, comprehensive components monitor as supported and complete, and all other monitors as not supported.

The external test equipment requests the following PID from the vehicle:

— PID 19_{16}: Bank 2 - Sensor 2, PID is supported by ECU#1 (ECM).

Table 32 — Request current powertrain diagnostic 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 current powertrain diagnostic data request SID	01_{16}	SIDRQ
#2	PID: Oxygen Sensor Output Voltage (B2 - S2) Short-term Fuel Trim (B2 - S2)	19_{16}	PID

Table 33 — Request current powertrain diagnostic 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 current powertrain diagnostic data response SID	41_{16}	SIDPR
#2	PID: Oxygen Sensor Output Voltage (B2 - S2) Short-term Fuel Trim (B2 - S2)	19_{16}	PID
#3	Oxygen Sensor Output Voltage (B2 - S2): 0,8 Volt	$A0_{16}$	DATA_A
#4	Short-term Fuel Trim (B2 - S2): 93,7 %	78_{16}	DATA_B

NOTE ECU#2 does not support PID 19_{16} and therefore does not send a response message.

7.2 Service 02_{16} — Request powertrain freeze frame data

7.2.1 Functional description

The purpose of this service is to allow access to emission-related data values in a freeze frame. This allows expansion to meet manufacturer-specific requirements not necessarily related to the required freeze frame and not necessarily containing the same data values as the required freeze frame. The request message includes a parameter identification (PID) value that indicates to the on-board system the specific information requested. PID specifications, scaling information, and display formats for the freeze frame are included in SAE J1979-DA.

The ECU(s) shall respond to this message by transmitting the requested data value stored by the system. All data values returned for sensor readings will be actual stored readings, not default or substitute values used by the system because of a fault with that sensor.

Not all PIDs are applicable or supported by all systems. PID 00_{16} is a bit-encoded PID that indicates, for each ECU, which PIDs that ECU supports. Therefore, PID 00_{16} shall be supported by all ECUs that respond to a Service 02_{16} request as specified, even if the ECU does not have a freeze frame stored at the time of the request.

SAE J1979-DA defines how to encode supported PIDs.

PID 02_{16} indicates the DTC that caused the freeze frame data to be stored. If freeze frame data are not stored in the ECU, the system shall report 0000_{16} as the DTC. Any data reported when the stored DTC is 0000_{16} may not be valid.

The frame number byte shall indicate 00_{16} for the mandated freeze frame data. Manufacturers may optionally save additional freeze frames and use this service to obtain that data by specifying the freeze frame number in the request message. If a manufacturer uses these additional freeze frames, they will be stored under conditions specified by the manufacturer and contain data specified by the manufacturer.

7.2.2 Message data bytes

7.2.2.1 Request powertrain freeze frame data request message definition (read-supported PIDs)

Table 34 — Request powertrain freeze frame data request message (read-supported PIDs)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request powertrain freeze frame data request SID	M	02_{16}	SIDRQ
#2	PID (see SAE J1979-DA)	M	XX_{16}	PID
#3	frame #	M	XX_{16}	FRNO

7.2.2.2 Request powertrain freeze frame data response message definition (report supported PIDs)

Table 35 — Request powertrain freeze frame data response message (report supported PIDs)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request powertrain freeze frame data response SID	M	42_{16}	SIDPR
#2	PID	M	XX_{16}	PID
#3	frame #	M	XX_{16}	FRNO
	data record of supported PIDs = [			DATAREC_
#4	Data A: supported PIDs,	M	XX_{16}	DATA_A
#5	Data B: supported PIDs,	M	XX_{16}	DATA_B
#6	Data C: supported PIDs,	M	xXX_{16}	DATA_C
#7	Data D: supported PIDs]	M	XX_{16}	DATA_D

7.2.2.3 Request powertrain freeze frame data request message definition (read freeze frame PID value)

Table 36 — Request powertrain freeze frame data request message (read freeze frame PID value)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request current powertrain diagnostic data request SID	M	02_{16}	SIDRQ
#2	PID (see SAE J1979-DA)	M/C[a]	XX_{16}	PID
#3	frame #	M	XX_{16}	FRNO

[a] C = Conditional. PID value shall be one of the supported PIDs of previous response message.

7.2.2.4 Request powertrain freeze frame data response message definition (report freeze frame PID value)

Table 37 — Request powertrain freeze frame data response message (report freeze frame PID value)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request powertrain freeze frame data response SID	M	42_{16}	SIDPR
#2	PID	M	XX_{16}	PID
#3	frame #	M	XX_{16}	FRNO
	data record = [			DATAREC_
#4	Data A,	M	XX_{16}	DATA_A
#5	Data B,	C[a]	XX_{16}	DATA_B
#6	Data C,	C	XX_{16}	DATA_C
#7	Data D]	C	XX_{16}	DATA_D

[a] C = Conditional. Data B - D depend on selected PID value.

7.2.3 Parameter definition

7.2.3.1 PIDs supported

SAE J1979-DA specifies the interpretation of the data record of supported PIDs.

7.2.3.2 PID and data byte descriptions

SAE J1979-DA specifies standardized emission-related parameters.

7.2.3.3 Frame # description

The frame number identifies the freeze frame, which includes emission-related data values in case an emission-related DTC is detected by the ECU.

7.2.4 Message example

7.2.4.1 General

The example below shows how the “Request powertrain freeze frame data” service shall be implemented.

7.2.4.2 Step #1: Request supported powertrain freeze frame PIDs from vehicle

The external test equipment requests all supported powertrain freeze frame PIDs of freeze frame 00_{16} from the vehicle. Refer to the example of Service 01_{16} on how to request supported PIDs.

As a result of the supported PID request, the external test equipment creates an internal list of supported PIDs for each ECU. ECU#1 (ECM) supports the following PIDs: 02_{16} – 09_{16}, $0B_{16}$ – $0E_{16}$. ECU#2 (TCM) does not support any PIDs for this service.

7.2.4.3 Step #2: Request PID 02_{16} "DTC which caused freeze frame to be stored" from vehicle

7.2.4.3.1 Case #1: Freeze frame data are stored in ECU#1

Now the external test equipment requests PID 02_{16} of freeze frame 00_{16} from the vehicle. Since the ECU#2 (TCM) does not 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 02_{16} "DTC that caused required freeze frame data storage" is set to the DTC P0130.

Table 38 — 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	02_{16}	SIDRQ
#2	PID: DTC that caused required freeze frame data storage	02_{16}	PID
#3	Frame #	00_{16}	FRNO

Table 39 — 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	42_{16}	SIDPR
#2	PID: DTC that caused required freeze frame data storage	02_{16}	PID
#3	Frame #: 00	00_{16}	FRNO
#4	DTC High Byte of P0130	01_{16}	DATA_A
#5	DTC Low Byte of P0130	30_{16}	DATA_B

7.2.4.3.2 Case #2: No freeze frame data are stored in any ECU

If no freeze frame data are stored, then the ECU(s) which support(s) this service but does (do) not have any freeze frame stored shall send a response message with the parameter values of DATA_A and DATA_B of PID 02_{16} "DTC that caused required freeze frame data storage" set to 0000_{16}.

Table 40 — 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	02_{16}	SIDRQ
#2	PID: DTC that caused required freeze frame data storage	02_{16}	PID
#3	Frame #: 00	00_{16}	FRNO

Table 41 — Request powertrain freeze frame data response message (Service 02_{16}, PID 02_{16}, Frame # 00_{16})

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	42_{16}	SIDPR
#2	PID: DTC that caused required freeze frame data storage	02_{16}	PID
#3	Frame #: 00	00_{16}	FRNO
#4	DTC High Byte: zero value indicates that no freeze frame is stored	00_{16}	DATA_A
#5	DTC Low Byte: zero value indicates that no freeze frame is stored	00_{16}	DATA_B

NOTE The DTC value reported is 0000_{16}, therefore no valid freeze frame data are stored for supported PIDs.

7.3 Service 03_{16} — Request emission-related diagnostic trouble codes

7.3.1 Functional description

The purpose of this service is to enable the external test equipment to obtain "confirmed" emission-related DTCs. This shall be a two-step process for the external test equipment:

— Step 1: Send a Service 01_{16}, PID 01_{16} request to get the number of emission-related DTCs from all ECUs that have this available. Each ECU that has a DTC(s) stored will respond with a message that includes the number of stored codes to be reported. If an ECU that is capable of storing emission-related DTCs does not have stored DTCs, then that ECU shall respond with a message indicating zero (0) DTCs are stored;

— Step 2: Send a Service 03_{16} request for all emission-related DTCs. Each ECU that has DTCs will respond with one or more messages, each containing up to three (3) DTCs. If no emission-related DTCs are stored in the ECU, then the ECU may not respond to this request.

If additional DTCs are set between the time that the number of DTCs is reported by an ECU and the DTCs are reported by an ECU, then the number of DTCs reported could exceed the number expected by the external test equipment. In this case, the external test equipment shall repeat this cycle until the number of DTCs reported equals the number expected based on the Service 01_{16}, PID 01_{16} response.

DTCs are transmitted in two (2) bytes of information for each DTC. The first two (2) bits (high order) of the first (1) byte for each DTC indicate whether the DTC is a powertrain, chassis, body, or network DTC (refer to SAE J2012 for additional interpretation of this structure). The second two (2) bits shall indicate the first (1) digit of the DTC (0 through 3). The second (2) nibble of the first (1) byte and the entire second (2) byte are the next three (3) hexadecimal characters of the actual DTC reported hexadecimal. A powertrain DTC transmitted as 0143_{16} shall be displayed as P0143 (see Figure 17).

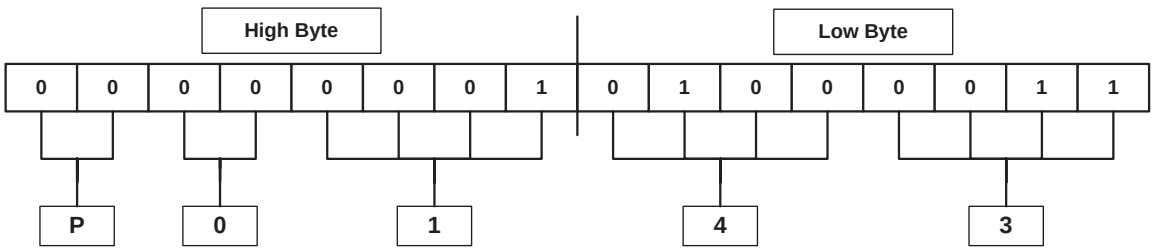


Figure 17 — Diagnostic trouble code encoding example DTC P0143

If fewer than three (3) DTCs are reported, the response message used to report DTCs shall have their unused bytes set to zero (0) to maintain the required fixed message length for all messages. If there

are no DTCs to report, a response message is allowed but not required for SAE J1850 and ISO 9141-2 interfaces. For ISO 14230-4 interfaces, the ECU will respond with a report containing no DTCs (DTC#1, DTC#2, and DTC#3 shall be all set to 00_{16}).

7.3.2 Message data bytes

7.3.2.1 Request current powertrain diagnostic data request message definition (PID 01_{16})

Table 42 — Request current powertrain diagnostic data request message (PID 01_{16})

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request current powertrain diagnostic data request SID	M	01_{16}	SIDRQ
#2	PID {Number of emission-related DTCs and MIL status}	M	01_{16}	PID

7.3.2.2 Request current powertrain diagnostic data response message definition (PID 01_{16})

Table 43 — Request current powertrain diagnostic data response message (PID 01_{16})

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request current powertrain diagnostic data response SID	M	41_{16}	SIDPR
#2	PID {number of emission-related DTCs and MIL status}	M	01_{16}	PID
	data record = [			DATAREC_
#3	Data A	M	XX_{16}	DATA_A
#4	Data B,	M	XX_{16}	DATA_B
#5	Data C,	M	XX_{16}	DATA_C
#6	Data D]	M	XX_{16}	DATA_D

7.3.2.3 Request emission-related DTC request message definition

Table 44 — Request emission-related DTC request message

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request emission-related DTC request SID	M	03_{16}	SIDRQ

7.3.2.4 Request emission-related DTC response message definition

Table 45 — Request emission-related DTC response message

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request emission-related DTC response SID	M	43_{16}	SIDPR
#2 #3	DTC#1 (High Byte) DTC#1 (Low Byte)	M/C[a] M/C	XX_{16} XX_{16}	DTC1HI DTC1LO
#4 #5	DTC#2 (High Byte) DTC#2 (Low Byte)	M/C M/C	XX_{16} XX_{16}	DTC2HI DTC2LO
#6 #7	DTC#3 (High Byte) DTC#3 (Low Byte)	M/C M/C	XX_{16} XX_{16}	DTC3HI DTC3LO

[a] C = Conditional. DTC#1, DTC#2, and DTC#3 are always present. If no valid DTC number is included, the DTC values shall contain 00_{16}.

7.3.3 Parameter definition

This service does not support any parameters.

7.3.4 Message example

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 vehicle supports the ISO 14230-4 protocol. 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 DTC stored (response message is optional for ISO 9141-2 and SAE J1850)

The external test equipment requests the following PID from the vehicle:

— PID 01_{16}: Number of emission-related DTCs and MIL status, PID is supported by ECU#1 (ECM), ECU#2 (TCM), and ECU#3 (ABS/Traction Control)

Table 46 — Request current powertrain diagnostic 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 current powertrain diagnostic data request SID	01_{16}	SIDRQ
#2	PID: Number of emission-related DTCs and MIL status	01_{16}	PID

Table 47 — Request current powertrain diagnostic 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 current powertrain diagnostic data response SID	41_{16}	SIDPR
#2	PID: Number of emission-related DTCs and MIL status	01_{16}	PID
#3	MIL: ON; Number of emission-related DTCs: 06_{16}	86_{16}	DATA_A
#4	Misfire -, Fuel system -, Comprehensive monitoring	33_{16}	DATA_B
#5	Catalyst -, Heated catalyst -, …, monitoring supported	FF_{16}	DATA_C
#6	Catalyst -, Heated catalyst -, …, monitoring test complete/not complete	63_{16}	DATA_D

Table 48 — Request current powertrain diagnostic data response message

Message Direction:	ECU#2 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request current powertrain diagnostic data response SID	41_{16}	SIDPR
#2	PID: Number of emission-related DTCs and MIL status	01_{16}	PID
#3	MIL: OFF; Number of emission-related DTCs: 01_{16}	01_{16}	DATA_A

Table 48 *(continued)*

Message Direction:	ECU#2 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#4	Comprehensive monitoring: supported, test complete	44_{16}	DATA_B
#5	Catalyst -, Heated catalyst -, ..., monitoring supported	00_{16}	DATA_C
#6	Catalyst -, Heated catalyst -, ..., monitoring test complete/not complete	00_{16}	DATA_D

Table 49 — Request current powertrain diagnostic data response message

Message Direction:	ECU#3 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request current powertrain diagnostic data response SID	41_{16}	SIDPR
#2	PID: Number of emission-related DTCs and MIL status	01_{16}	PID
#3	MIL: OFF; Number of emission-related DTCs: 00_{16}	00_{16}	DATA_A
#4	Comprehensive monitoring: supported, test complete	00_{16}	DATA_B
#5	Catalyst -, Heated catalyst -, ..., monitoring supported	00_{16}	DATA_C
#6	Catalyst -, Heated catalyst -, ..., monitoring test complete/not complete	00_{16}	DATA_D

The external test equipment requests emission-related DTCs because ECU#1 has six (6) DTCs stored, ECU#2 has one (1) DTC stored, and ECU#3 has no (0) DTC stored.

Table 50 — 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**	**Mnemonic**
#1	Request emission-related DTC request SID	03_{16}	SIDRQ

Table 51 — 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**	**Mnemonic**
#1	Request emission-related DTC response SID	43_{16}	SIDPR
#2	DTC#1 High Byte of P0143	01_{16}	DTC1HI
#3	DTC#1 Low Byte of P0143	43_{16}	DTC1LO
#4	DTC#2 High Byte of P0196	01_{16}	DTC2HI
#5	DTC#2 Low Byte of P0196	96_{16}	DTC2LO
#6	DTC#3 High Byte of P0234	02_{16}	DTC3HI
#7	DTC#3 Low Byte of P0234	34_{16}	DTC3LO

Table 52 — 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**	**Mnemonic**
#1	Request emission-related DTC response SID	43_{16}	SIDPR
#2	DTC#1 High Byte of P0443	04_{16}	DTC1HI
#3	DTC#1 Low Byte of P0443	43_{16}	DTC1LO
#4	DTC#2 High Byte: 00_{16}	00_{16}	DTC2HI
#5	DTC#2 Low Byte: 00_{16}	00_{16}	DTC2LO
#6	DTC#3 High Byte: 00_{16}	00_{16}	DTC3HI
#7	DTC#3 Low Byte: 00_{16}	00_{16}	DTC3LO

Table 53 — 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**	**Mnemonic**
#1	Request emission-related DTC response SID	43_{16}	SIDPR
#2	DTC#1 High Byte of P02CD	02_{16}	DTC1HI
#3	DTC#1 Low Byte of P02CD	CD_{16}	DTC1LO
#4	DTC#2 High Byte of P0357	03_{16}	DTC2HI
#5	DTC#2 Low Byte of P0357	57_{16}	DTC2LO
#6	DTC#3 High Byte of P0A24	$0A_{16}$	DTC3HI
#7	DTC#3 Low Byte of P0A24	24_{16}	DTC3LO

Table 54 — 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**	**Mnemonic**
#1	Request emission-related DTC response SID	43_{16}	SIDPR
#2	DTC#1 High Byte: P0000	00_{16}	DTC1HI
#3	DTC#1 Low Byte: P0000	00_{16}	DTC1LO
#4	DTC#2 High Byte: P0000	00_{16}	DTC2HI
#5	DTC#2 Low Byte: P0000	00_{16}	DTC2LO
#6	DTC#3 High Byte: P0000	00_{16}	DTC3HI
#7	DTC#3 Low Byte: P0000	00_{16}	DTC3LO

For ISO 9141-2 and SAE J1850 protocols, the ECU#3 response message is optional because there is no DTC stored. If ISO 14230-4 protocol is supported by the vehicle, ECU#3 shall send a positive response message with no DTCs.

7.4 Service 04_{16} — Clear/reset emission-related diagnostic information

7.4.1 Functional description

The purpose of this service is to provide a means for the external test equipment to command ECUs to clear all emission-related diagnostic information. This includes the following:

— MIL and number of diagnostic trouble codes (can be read with Service 01_{16}, PID 01_{16});

— clear the I/M (inspection/maintenance) readiness bits (can be read with Service 01_{16}, PID 01_{16});

— confirmed diagnostic trouble codes (can be read with Service 03_{16});

— pending diagnostic trouble codes (can be read with Service 07_{16});

— diagnostic trouble code for freeze frame data (can be read with Service 02_{16}, PID 02_{16});

— freeze frame data (can be read with Service 02_{16});

— oxygen sensor test data (can be read with Service 05_{16});

— status of system monitoring tests (can be read with Service 01_{16}, PID 41_{16});

— on-board monitoring test results (can be read with Service 06_{16});

— distance travelled while MIL is activated (can be read with Service 01_{16}, PID 21_{16});

— number of warm-ups since DTCs cleared (can be read with Service 01_{16}, PID 30_{16});

— distance travelled since DTCs cleared (can be read with Service 01_{16}, PID 31_{16});

— engine run time while MIL is activated (can be read with Service 01_{16}, PID $4D_{16}$);

— engine run time time since DTCs cleared (can be read with Service 01_{16}, PID $4E_{16}$).

Other manufacturer-specific “clearing/resetting” actions may also occur in response to this request message. For safety and/or technical design reasons, some ECUs may not respond to this service under all conditions. All ECUs shall respond to this service request with the ignition ON and with the engine not running. ECUs that cannot perform this operation under other conditions, such as with the engine running, will ignore the request with SAE J1850 and ISO 9141-2 interfaces or will send a negative response message with ISO 14230-4 interfaces, as described in ISO 14230-4.

7.4.2 Message data bytes

7.4.2.1 Clear/reset emission-related diagnostic information request message definition

Table 55 — Clear/Reset emission-related diagnostic information request message

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Clear/reset emission-related diagnostic information request SID	M	04	SIDRQ

7.4.2.2 Clear/reset emission-related diagnostic information response message definition

Table 56 — Clear/Reset emission-related diagnostic information response message

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Clear/reset emission-related diagnostic information response SID	M	44	SIDPR

7.4.3 Parameter definition

This service does not support any parameters.

7.4.4 Message example

This example is based on the example of Service 03_{16} as described in 7.3.4. The external test equipment commands the vehicle to clear/reset emission-related diagnostic information with the engine running. The ECU#1 (ECM) and ECU#2 (TCM) will send a response message to confirm that all emission-related diagnostic information is cleared. For ISO 9141-2 and SAE J1850 protocols, ECU#3 (ABS/Traction control) will not send a response message because the conditions to perform the requested action are not met. For ISO 14230-4 protocol, ECU#3 will send a negative response message with NRC 22_{16} - conditionsNotCorrect. In such case the external test equipment shall post a message with "Stop engine and turn ON ignition" and then repeat the Service 04_{16} command and check for response messages from all emission-related ECUs installed in the vehicle.

Table 57 — 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**	**Mnemonic**
#1	Clear/reset emission-related diagnostic information request SID	04_{16}	SIDRQ

Table 58 — 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**	**Mnemonic**
#1	Clear/reset emission-related diagnostic information response SID	44_{16}	SIDPR

Table 59 — 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**	**Mnemonic**
#1	Clear/reset emission-related diagnostic information response SID	44_{16}	SIDPR

Table 60 — Negative response message

Message Direction:	ECU#3 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Negative Response Service Identifier	$7F_{16}$	SIDNR
#2	Clear/reset emission-related diagnostic information request SID	04_{16}	SIDRQ
#3	Negative Response Code: conditionsNotCorrect	22_{16}	NR_CNC

For ISO 14230-4 protocol, the conditions of ECU#3 to clear/reset emissions-related diagnostic information is not met. Therefore, ECU#3 sends a negative response message with NRC "conditionsNotCorrect". The external test equipment shall repeat the request after the conditions of the vehicle have changed by the user. Now, all ECUs shall send a positive response message to the external test equipment to confirm successful operation of the clear/reset emission-related diagnostic information service.

7.5 Service 05_{16} — Request oxygen sensor monitoring test results

7.5.1 Functional description

The purpose of this service is to allow access to the on-board oxygen sensor monitoring test results. The same information may be obtained by the use of Service 06_{16}.

The request message for test results includes a Test ID value that indicates the information requested. Test value definitions, scaling information, and display formats are included in SAE J1979-DA.

Many methods may be used to calculate test results for this service by different manufacturers. If data values are to be reported using these messages that are different from those specified, ranges of test values have been assigned that can be used which have standard units of measure. The external test equipment can convert these values and display them in the standard units.

The ECU shall respond to this message by transmitting the requested test data last determined by the system. The latest test results are to be retained, even over multiple ignition OFF cycles, until replaced by more recent test results. Test results are requested by Test ID.

Not all test values are applicable or supported by all vehicles. An optional feature of this service is for the ECU to indicate which Test IDs are supported. Test ID 00_{16} is a bit-encoded value that indicates support for Test IDs from 01_{16} to 20_{16}. Test ID 20_{16} indicates support for Test IDs 21_{16} through 40_{16}, etc. This is the same concept as used for PID support in Services 01_{16} and 02_{16} as specified in SAE J1979-DA. If Test ID 00_{16} is not supported, then the ECU does not use this feature to indicate Test ID support.

7.5.2 Message data bytes

7.5.2.1 Request oxygen sensor monitoring test results request message definition (read-supported TIDs)

Table 61 — Request oxygen sensor monitoring test results request message (read-supported TIDs)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request oxygen sensor monitoring test results request SID	M	05_{16}	SIDRQ
#2	Test ID (see SAE J1979-DA)	M	XX_{16}	TID
#3	O2 Sensor #	M	XX_{16}	O2SNO

Table 62 — Request oxygen sensor monitoring test results response message (report supported TIDs)

Data Byte	Parameter Name		Cvt	Byte Value	Mnemonic
#1	Request oxygen sensor monitoring test results response SID		M	45_{16}	SIDPR
#2	Test ID		M	XX_{16}	TID
#3	O2 Sensor #		M	XX_{16}	O2SNO
#4	data record of supported test IDs = [	Data A: supported Test IDs,	M	XX_{16}	DATA_A
#5		Data B: supported Test IDs,	M	XX_{16}	DATA_B
#6		Data C: supported Test IDs,	M	XX_{16}	DATA_C
#7		Data D: supported Test IDs]	M	XX_{16}	DATA_D

7.5.2.2 Request oxygen sensor monitoring test results request message definition (read TID values)

Table 63 — Request oxygen sensor monitoring test results request message (read TID values)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request oxygen sensor monitoring test results request SID	M	05_{16}	SIDRQ
#2	Test ID	M	XX_{16}	TID
#3	O2 Sensor #	M	XX_{16}	O2SNO

7.5.2.3 Request oxygen sensor monitoring test results response message definition (report TID values)

Table 64 — Request oxygen sensor monitoring test results response message (report TID values)

Data Byte	Parameter Name		Cvt	Byte Value	Mnemonic
#1	Request oxygen sensor monitoring test results response SID		M	45_{16}	SIDPR
#2	Test ID		M	XX_{16}	TID
#3	O2 Sensor #		M	XX_{16}	O2SNO
#4	data record of Test ID = [	Test Value	M	XX_{16}	TESTVAL
#5		Minimum Limit	C[a]	XX_{16}	MINLIMIT
#6		Maximum Limit]	C	XX_{16}	MAXLIMIT

[a] C = Conditional. If the supported test ID is a constant (01_{16} – 04_{16}), the parameters Minimum and Maximum Limit shall not be included.

7.5.3 Parameter definition

7.5.3.1 Test IDs supported

The Test IDs supported is the same concept as used for PID support in Services 01_{16} and 02_{16} as specified in SAE J1979-DA.

7.5.3.2 Test ID and data byte descriptions

SAE J1979-DA specifies standardized and vehicle manufacturer specific Test ID ranges.

7.5.3.3 Oxygen sensor location definition

The oxygen sensor location value used in the request message shall indicate the oxygen sensor location as defined by PID 13_{16} or $1D_{16}$ as specified in SAE J1979-DA.

Table 65 — Oxygen sensor location description

Oxygen sensor location (one, and only one bit can be set to a 1)		
Bit	**Sensor location[a]**	**Alternative sensor location[b]**
0	Bank 1 - Sensor 1	Bank 1 - Sensor 1
1	Bank 1 - Sensor 2	Bank 1 - Sensor 2
2	Bank 1 - Sensor 3	Bank 2 - Sensor 1
3	Bank 1 - Sensor 4	Bank 2 - Sensor 2
4	Bank 2 - Sensor 1	Bank 3 - Sensor 1
5	Bank 2 - Sensor 2	Bank 3 - Sensor 2
6	Bank 2 - Sensor 3	Bank 4 - Sensor 1
7	Bank 2 - Sensor 4	Bank 4 - Sensor 2

[a] If Service 01_{16} PID 13_{16} supported.

[b] If Service 01_{16} PID $1D_{16}$ supported.

7.5.3.4 Test result description

Table 66 — Test result description

Hex	# of bytes	Description
00_{16} – FF_{16}	1	The test result parameter includes either a constant or a calculated value depending on the Test ID.

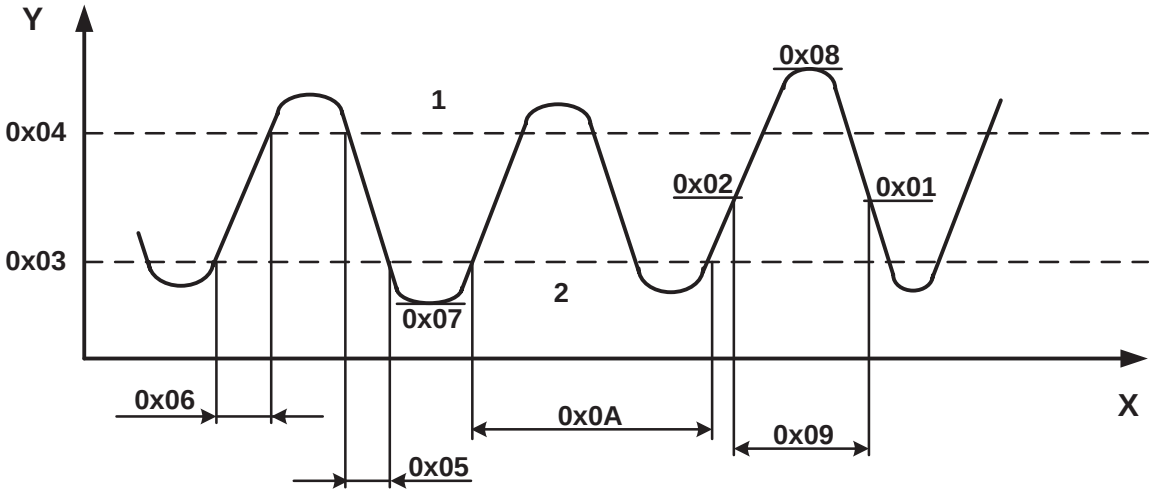
7.5.3.5 Minimum and maximum test limit description

The minimum and maximum test limit description shown in Table 67 defines the test limit value which is either a minimum or a maximum value to which the test results are compared. The test limit is a one-byte unsigned numeric value (0 - 255).

Table 67 — Minimum and maximum test limit description

Test Limit	# of bytes	Description
Minimum	1	The minimum test limit (only for calculated test result) is the minimum value to which the test result is compared.
Maximum	1	The maximum test limit (only for calculated test result) is the maximum value to which the test result is compared.

For results of latest mandated on-board oxygen sensor monitoring test, see Figure 18.



Key

1 rich

2 lean

Figure 18 — Test ID value example

7.5.4 Message example

The example below shows how the "Request oxygen sensor monitoring test results" service shall be implemented.

7.5.4.1 Step #1: Request oxygen sensor monitoring test results (request for supported Test IDs) from vehicle

The external test equipment requests all supported Test IDs from the vehicle. Refer to the example of Service 01_{16} for how to request supported PIDs (same concept is used for supported TIDs). PID 13_{16} is supported by ECU#1. This is important information for the external test equipment in order to identify the correct O2 sensor location.

As a result of the supported TID request, the external test equipment creates an internal list of supported TIDs for each ECU: The ECU#1 (ECM) supports Test IDs 01_{16} - 06_{16}, 70_{16}, 71_{16}, and 81_{16}. The ECU#2 (TCM) does not support any Test IDs.

7.5.4.2 Step #2: Request oxygen sensor monitoring test results from vehicle

The external test equipment sends two (2) "Request oxygen sensor monitoring test results" request messages to the vehicle. The two (2) request messages include the following Test IDs:

— 1st request message: Test IDs 01_{16};

— 2nd request message: Test IDs 05_{16}.

In general, the external test equipment should read the test status of Service 01_{16} PID 01_{16} prior to execution Service 05_{16} with Test ID 01_{16} and 05_{16} to verify whether the tests are supported and completed. The test values reported may be invalid if the test is not completed.

Table 68 — 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**	**Mnemonic**
#1	Request oxygen sensor monitoring test results request SID	05_{16}	SIDRQ
#2	TID: Rich to lean sensor threshold voltage (constant)	01_{16}	TID
#3	O2 Sensor #: Bank 1 - Sensor 1	01_{16}	O2SNO

Table 69 — 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**	**Mnemonic**
#1	Request oxygen sensor monitoring test results response SID	45_{16}	SIDPR
#2	TID: Rich to lean sensor threshold voltage (constant)	01_{16}	TID
#3	O2 Sensor #: Bank 1 - Sensor 1	01_{16}	O2SNO
#4	Test Limit: 450 mV	$5A_{16}$	TESTVAL

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

Table 70 — 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**	**Mnemonic**
#1	Request oxygen sensor monitoring test results request SID	05_{16}	SIDRQ
#2	TID: Rich to lean sensor switch time (calculated)	05_{16}	TID
#3	O2 Sensor #: Bank 1 - Sensor 1	01_{16}	O2SNO

Table 71 — 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**	**Mnemonic**
#1	Request oxygen sensor monitoring test results response SID	45_{16}	SIDPR
#2	TID: Rich to lean sensor switch time (calculated)	05_{16}	TID
#3	O2 Sensor #: Bank 1 - Sensor 1	01_{16}	O2SNO
#4	Test Limit: 72 ms (milliseconds)	12_{16}	TESTVAL
#5	Minimum Limit: 0 ms	00_{16}	MINLIMIT
#6	Maximum Limit: 100 ms	19_{16}	MAXLIMIT

7.6 Service 06_{16} — Request On-board monitoring test results for specific monitored systems

7.6.1 Functional description

The purpose of this service is to allow access to the results of On-Board Diagnostic monitoring tests for specific components/systems. Examples are catalyst monitoring and the evaporative system monitoring.

The vehicle manufacturer is responsible for assigning Test IDs and Component IDs for tests of different systems and components. The latest valid test results are to be retained, even over multiple ignition OFF cycles, until replaced by more recent test results. Test results are requested by Test ID. Test results are reported only for supported combinations of test limit type and component ID, and are reported as positive (unsigned) values. Only one test limit is included in a response message, but that limit could be either a minimum or a maximum limit. If both a minimum and maximum test limit are to be reported, then two (2) response messages will be transmitted, in any order. The most significant bit (MSB) of the "test limit type/component ID" byte will be used to indicate the test limit type.

A feature of this service is for the ECU to indicate which Test IDs are supported. Test ID 00_{16} is a bit-encoded value that indicates support for Test IDs from 01_{16} to 20_{16}. Test ID 20_{16} indicates support for Test IDs 21_{16} through 40_{16}, etc. This is the same concept as used for PID support in Services 01_{16} and 02_{16} as specified in SAE J1979-DA.

This service can be used as an alternative to Service 05_{16} to report oxygen sensor test results.

A unique method shall be utilized for displaying data for monitors that have multiple tests. Many OBD monitors have multiple tests that are done in either a serial or parallel manner. If a monitor uses multiple Test ID/Component ID combinations that may not all complete at the same time, the following method shall be used to update the stored test results at the time of monitor completion.

After the monitor completes, update all Test ID/Component ID combinations (or "test results") that were utilized by the monitor with appropriate passing or failing results. If a test result (or "Test ID/Component ID") was not utilized during this monitoring event, set the Test Values and Minimum and Maximum Test Limits to their initial values (test not completed). Test results from the previously completed monitoring events shall not be mixed with test results from the current completed monitoring event.

In some cases, test results (or "Test ID/Component ID combinations") will be displayed as being incomplete even though the monitor (as indicated by PID 41_{16}) was successfully completed and either passed or failed. In other cases, some Test IDs will show passing results while others will show failing results after the monitor (as indicated by PID 41_{16}) was successfully completed and failed. Note that OBD-II regulations prohibit a passing monitor from showing any failing test results. If an initial serial test indicates a failure and a subsequent re-test of the system indicates a passing result, the test that was utilized to make the passing determination should be displayed while the failing test that was utilized to make the initial determination should be reset to its initial values (test not completed).

As an example of a serial monitor, an evaporative system monitor can fail for a large evaporative system leak and never continue to test for small leaks or very small leaks. In this case, the Component ID for the large leak would show a failing result, while the small leak test and the very small leak test would show incomplete. As an example of the parallel monitor, a purge valve flow monitor can pass by having a large rich lambda shift, a large lean lambda shift, or a large engine rpm increase. If the purge valve is activated and a large rich lambda shift occurs, the Component ID for the rich lambda shift would show a passing result while the other two Component IDs would show incomplete. Since some Component IDs for a completed monitor will show incomplete, PID 41_{16} shall be used to determine monitor completion status.

7.6.2 Message data bytes

7.6.2.1 Request on-board monitoring test results for specific monitored systems request message definition (read-supported TIDs)

Table 72 — Request on-board monitoring test results for specific monitored systems request message (read-supported TIDs)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request on-board monitoring test results for specific monitored systems request SID	M	06_{16}	SIDRQ
#2	Test ID (see SAE J1979-DA)	M	XX_{16}	TID

7.6.2.2 Request on-board monitoring test results for specific monitored systems response message definition (report supported TIDs)

Table 73 — Request on-board monitoring test results for specific monitored systems response message (report supported TIDs)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request on-board monitoring test results for specific monitored systems response SID	M	46_{16}	SIDPR
#2	Test ID	M	XX_{16}	TID
#3	Filler Byte	M	FF_{16}	FB
	data record of supported Test IDs = [			DATAREC_
#4	Data A: supported Test IDs,	M	XX_{16}	DATA_A
#5	Data B: supported Test IDs,	M	XX_{16}	DATA_B
#6	Data C: supported Test IDs,	M	XX_{16}	DATA_C
#7	Data D: supported Test IDs]	M	XX_{16}	DATA_D

7.6.2.3 Request on-board monitoring test results for specific monitored systems request message definition (read test results)

Table 74 — Request on-board monitoring test results for specific monitored systems request message (read test results)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request on-board monitoring test results for specific monitored systems request SID	M	06_{16}	SIDRQ
#2	Test ID (request test results)	M	XX_{16}	TID

7.6.2.4 Request on-board monitoring test results for Specific monitored systems response message definition (report test results)

Table 75 — Request on-board monitoring test results for specific monitored systems response message (report test results)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request on-board monitoring test results for specific monitored systems response SID	M	46_{16}	SIDPR
#2	Test ID (report test results)	M	XX_{16}	TID
#3	Test Limit Type and Component ID	M	XX_{16}	TLTCID
	data record of supported Test IDs = [			TIDREC_
#4	Test Value (High Byte)	M	XX_{16}	TVHI
#5	Test Value (Low Byte)	M	XX_{16}	TVLO
#6	Test Limit (High Byte)	C[a]	XX_{16}	TLHI
#7	Test Limit (Low Byte)]	C	XX_{16}	TLLO

[a] C = Conditional. If Test Limit is either a Minimum or a Maximum Limit depends on the parameter Test Limit Type and Component ID value (bit 7).

7.6.3 Parameter definition

7.6.3.1 Test IDs supported

The Test IDs supported is the same concept as used for PID support in Services 01_{16} and 02_{16} as specified in SAE J1979-DA.

7.6.3.2 Test ID and data byte descriptions

SAE J1979-DA specifies standardized and vehicle manufacturer-specific Test ID ranges, which are permitted to be supported in this service.

NOTE For ISO 9141-2, SAE J1850, and ISO 14230-4 protocols that SAE J1979-DA is recommended but not required. This is for backward compatibility and only applies to Test ID range 01_{16} – $1F_{16}$.

7.6.3.3 Test Limit Type and Component ID description

The Test Limit Type and Component ID is a one (1) byte parameter and are defined in Table 76.

Table 76 — Test Limit Type and Component ID description

Parameter Name	Bit	Description
Component ID	0 - 6	Component ID - manufacturer specified - necessary when multiple components or systems are present on the vehicle and have the same definition of Test ID. If the same test is performed on more than one component, multiple test results shall be reported for that Test ID. For example, a test for bank 1 catalyst can be the same as a test for a bank 2 catalyst, or a test for a pre-catalyst oxygen sensor can be the same as a test for a post-catalyst oxygen sensor. In either case, a request for a single Test ID would result in two test results being reported with different Component IDs.
Test Limit Type	7	Most significant bit (MSB) indicates type of test limit, where 0 - test limit is maximum value - test fails if test value is greater than this value, and 1 - test limit is minimum value - test fails if test value is less than this value.

7.6.3.4 Test Result description

The Test Result represents the test result and is defined in Table 77.

Table 77 — Test Result description

Parameter Name	# of Bytes	Description
Test Result	2 (High and Low Byte)	Test Result - this value shall be less than or equal to the test limit if MSB of Test Limit Type and Component ID byte is "0", and shall be greater than or equal to the test limit if MSB of Test Limit Type and Component ID byte is "1". The Test Value is a two-byte unsigned numeric value (0 – 65 535).

7.6.3.5 Test Limit description

The Test Limit is defined in Table 78.

Table 78 — Test Limit description

Parameter Name	# of Bytes	Description
Test Limit	2 (High and Low Byte)	The Test Limit value is either a minimum or a maximum value to which the test results are compared. The Test Limit is a two-byte unsigned numeric value (0 – 65 535).

7.6.4 Message example

Tables 79 to 81 below show how the "request on-board monitoring test results for specific monitored systems" service shall be implemented.

7.6.4.1 Step #1: Request on-board monitoring test results for specific monitored systems (request for supported Test IDs)

The external test equipment requests all supported Test IDs from the vehicle. Refer to the example of Service 01_{16} for guidance on requesting supported PIDs (the same concept is used for supported TIDs).

As a result of the supported TID request, the external test equipment creates an internal list of supported TIDs for each ECU. ECU#1 (ECM) supports Test ID 02. ECU#2 (TCM) does not support any Test IDs.

7.6.4.2 Step #2: Request on-board monitoring test results for specific monitored systems

The external test equipment sends a "request on-board monitoring test results for specific monitored systems" request message with one (1) supported Test ID to the vehicle. The response messages indicate which Component IDs are supported. The request message includes the following Test ID:

Test ID 02_{16} - Lean to rich sensor threshold voltage (constant), (supported Component IDs: 04_{16}, 16_{16}).

In general, the external test equipment should read the test status of Service 01_{16} PID 01_{16} prior to executing Service 06_{16} with Test ID 01_{16} and 06_{16} to verify whether the tests are supported and completed. The test values reported may be invalid if the test is not completed.

Table 79 — Request on-board monitoring test results for specific monitored systems 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 on-board monitoring test results for specific monitored systems request SID	06_{16}	SIDRQ
#2	TID Lean to rich sensor threshold voltage (constant)	02_{16}	TID

Table 80 — Request on-board monitoring test results for specific monitored systems 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 on-board monitoring test results for specific monitored systems response SID	46_{16}	SIDPR
#2	TID Lean to rich sensor threshold voltage (constant)	02_{16}	TID
#3	Test Limit Type: test limit is minimum value; Component ID: 04	84_{16}	TLTCID
#4	Test Value High Byte: test fails if test value is less than test limit	00_{16}	TVHI
#5	Test Value Low Byte: test fails if test value is less than test limit	10_{16}	TVLO
#6	Minimum Test Limit High Byte	00_{16}	TLHI
#7	Minimum Test Limit Low Byte	00_{16}	TLLO

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

Table 81 — Request on-board monitoring test results for specific monitored systems 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 on-board monitoring test results for specific monitored systems response SID	46_{16}	SIDPR
#2	TID Lean to rich sensor threshold voltage (constant)	02_{16}	TID
#3	Test Limit Type: test limit is maximum value; Component ID: 16_{16}	16_{16}	TLTCID
#4	Test Value High Byte: test fails if test value is greater than test limit	00_{16}	TVHI
#5	Test Value Low Byte: test fails if test value is greater than test limit	32_{16}	TVLO
#6	Maximum Test Limit High Byte	00_{16}	TLHI
#7	Maximum Test Limit Low Byte	20_{16}	TLLO

NOTE The above example shows that the test in ECU#1 for Test ID 02_{16} and Component ID 04_{16} passed and that the test in ECU#1 for Test ID 02_{16} and Component ID 16_{16} failed.

7.7 Service 07_{16} — Request emission-related diagnostic trouble codes detected during current or last completed driving cycle

7.7.1 Functional description

The purpose of this service is to enable the external test equipment to obtain "pending" diagnostic trouble codes detected during current or last completed driving cycle for emission-related components/systems. Service 07_{16} is required for all DTCs and is independent of Service 03_{16}. The intended use of this data is to assist the service technician after a vehicle repair and after clearing diagnostic information by reporting test results after a single driving cycle. If the test failed during the driving cycle, the DTC associated with that test will be reported. Test results reported by this service do not necessarily indicate a faulty component/system. If test results indicate a failure after additional driving, then the MIL will be illuminated and a DTC will be set and reported with Service 03_{16}, indicating a faulty component/system. This service can always be used to request the results of the latest test, independent of the setting of a DTC.

Test results for these components/systems are reported in the same format as the DTCs in Service 03_{16} (see the functional description for Service 03_{16}).

If fewer than three (3) DTC values are reported for failed tests, the response messages used to report the test results shall be filled with 00_{16} to fill seven (7) data bytes. This maintains the required fixed message length for all messages.

If there is no test failure to report, responses are permitted but not required for SAE J1850 and ISO 9141-2 interfaces. For ISO 14230-4 interfaces, the ECU will respond with a report containing no codes (all DTC values shall contain 00_{16}).

7.7.2 Message data bytes

7.7.2.1 Request emission-related diagnostic trouble codes detected during current or last completed driving cycle request message definition

Table 82 — Request emission-related diagnostic trouble codes detected during current or last completed driving cycle request message

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request emission-related diagnostic trouble codes detected during current or last completed driving cycle request SID	M	07_{16}	SIDRQ

7.7.2.2 Request emission-related diagnostic trouble codes detected during current or last completed driving cycle response message definition

Table 83 — Request emission-related diagnostic trouble codes detected during current or last completed driving cycle response message

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request emission-related diagnostic trouble codes detected during current or last completed driving cycle response SID	M	47_{16}	SIDPR
#2 #3	DTC#1 (High Byte) DTC#1 (Low Byte)	M/C[a] M/C	XX_{16} XX_{16}	DTC1HI DTC1LO
#4 #5	DTC#2 (High Byte) DTC#2 (Low Byte)	M/C M/C	XX_{16} XX_{16}	DTC2HI DTC2LO
#6 #7	DTC#3 (High Byte) DTC#3 (Low Byte)	M/C M/C	XX_{16} XX_{16}	DTC3HI DTC3LO

[a] C = Conditional. DTC#1, DTC#2, and DTC#3 are always present. If no valid DTC number is included, the DTC values shall contain 00_{16}.

7.7.3 Parameter definition

This service does not support any parameters.

7.7.4 Message example

Refer to message example of Service 03_{16}.

7.8 Service 08_{16} — Request control of on-board system, test, or component

7.8.1 Functional description

The purpose of this service is to enable the external test equipment to control the operation of an on-board system, test, or component.

The data bytes will be specified, if necessary, for each Test ID in SAE J1979-DA and will be unique for each Test ID. If any data bytes are unused for any test, they shall be filled with 00_{16} to maintain a fixed message length.

Possible uses for these data bytes in the request message are to

— turn on-board system/test/component on,

— turn on-board system/test/component off, and

— cycle on-board system/test/component for "*n*" seconds.

Possible uses for these data bytes in the response message are to

— report system status, and

— report test results.

A feature of this service is for the ECU to indicate which Test IDs are supported. Test ID 00_{16} is a bit-encoded value that indicates support for Test IDs from 01_{16} to 20_{16}. Test ID 20_{16} indicates support for Test IDs 21_{16} through 40_{16}, etc. This is the same concept as used for PID support in Services 01_{16} and 02_{16} as specified in SAE J1979-DA.

7.8.2 Message data bytes

7.8.2.1 Request control of on-board device request message definition (read-supported TIDs)

Table 84 — Request control of on-board device request message (read-supported TIDs)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request control of on-board device request SID	M	08_{16}	SIDRQ
#2	Test ID (see SAE J1979-DA)	M	XX_{16}	TID
	data record of Test ID = [			TIDREC_
#3	Data A,	M	00_{16}	DATA_A
#4	Data B,	M	00_{16}	DATA_B
#5	Data C,	M	00_{16}	DATA_C
#6	Data D,	M	00_{16}	DATA_D
#7	Data E]	M	00_{16}	DATA_E

7.8.2.2 Request control of on-board device response message definition (report supported TIDs)

Table 85 — Request control of on-board device response message (report supported TIDs)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request control of on-board device response SID	M	48_{16}	SIDPR
#2	Test ID	M	XX_{16}	TID
#3	Filler Byte	M	00_{16}	FB
	data record of supported Test IDs = [			TIDREC_
#4	Data A: supported Test IDs,	M	XX_{16}	DATA_A
#5	Data B: supported Test IDs,	M	XX_{16}	DATA_B
#6	Data C: supported Test IDs,	M	xXX_{16}	DATA_C
#7	Data D: supported Test IDs]	M	XX_{16}	DATA_D

7.8.2.3 Request control of on-board device request message definition (read TID values)

Table 86 — Request control of on-board device request message (read TID values)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request control of on-board device request SID	M	08_{16}	SIDRQ
#2	Test ID (request Test ID values)	M	XX_{16}	TID
	data record of Test ID = [			TIDREC_
#3	Data A,	M/C[a]	XX_{16}	DATA_A
#4	Data B,	M/C	XX_{16}	DATA_B
#5	Data C,	M/C	XX_{16}	DATA_C
#6	Data D,	M/C	XX_{16}	DATA_D
#7	Data E]	M/C	XX_{16}	DATA_E

[a] C = Conditional. Data A to E shall be filled with 00_{16} if unused.

7.8.2.4 Request control of on-board device response message definition (report TID values)

Table 87 — Request control of on-board device response message (report TID values)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request control of on-board device request SID	M	48_{16}	SIDRQ
#2	Test ID (request Test ID values)	M	XX_{16}	TID
	data record of Test ID = [			TIDREC_
#3	Data A,	M/C[a]	XX_{16}	DATA_A
#4	Data B,	M/C	XX_{16}	DATA_B
#5	Data C,	M/C	XX_{16}	DATA_C
#6	Data D,	M/C	XX_{16}	DATA_D
7	Data E]	M/C	XX_{16}	DATA_E

[a] C = Conditional. Data A to E shall be filled with 00_{16} if unused.

7.8.3 Parameter definition

7.8.3.1 Test IDs supported

Refer to SAE J1979-DA.

7.8.3.2 Test ID and data byte descriptions

Refer to SAE J1979-DA.

7.8.4 Message example

Tables 88 and 89 show how "request control of on-board system, test, or component" service shall be implemented.

7.8.4.1 Step #1: Request control of on-board system, test, or component (request for supported Test IDs)

The external test equipment requests all supported Test IDs from the vehicle. Refer to the example of Service 01_{16} for guidance on requesting supported Test IDs (the same concept is used for supported TIDs).

As a result of the supported TID request, the external test equipment creates an internal list of supported PIDs for each ECU. ECU#1 (ECM) supports Test ID 01_{16}. ECU#2 (TCM) does not support any Test IDs and therefore does not send a response message.

7.8.4.2 Step #2: Request control of on-board device (Service 08_{16}, Test ID 01_{16})

The external test equipment sends a "request control of on-board device" message with one (1) supported Test ID 01_{16} to the vehicle.

Table 88 — 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**	**Mnemonic**
#1	Request control of on-board device request SID	08_{16}	SIDRQ
#2	TID: Evaporative system leak test	01_{16}	TID
#3	Data A: 00_{16}	00_{16}	DATA_A
#4	Data B: 00_{16}	00_{16}	DATA_B
#5	Data C: 00_{16}	00_{16}	DATA_C
#6	Data D: 00_{16}	00_{16}	DATA_D
#7	Data E: 00_{16}	00_{16}	DATA_E

Table 89 — 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**	**Mnemonic**
#1	Request control of on-board device response SID	48_{16}	SIDPR
#2	TID: Evaporative system leak test	01_{16}	TID
#3	Data A: 00_{16}	00_{16}	DATA_A
#4	Data B: 00_{16}	00_{16}	DATA_B
#5	Data C: 00_{16}	00_{16}	DATA_C
#6	Data D: 00_{16}	00_{16}	DATA_D
#7	Data E: 00_{16}	00_{16}	DATA_E

NOTE ECU#2 does not support the Test ID and therefore does not send a response message.

7.9 Service 09_{16} — Request vehicle information

7.9.1 Functional description

The purpose of this service is to enable the external test equipment to request vehicle-specific vehicle information such as Vehicle Identification Number (VIN) and Calibration IDs. Some of this information may be required by regulations and some should be reported in a standard format if supported by the vehicle manufacturer. INFOTYPEs are defined in SAE J1979-DA.

A feature of this service is for the ECU to indicate which INFOTYPEs are supported (support of INFOTYPE 00_{16} is required for ISO 9141-2). INFOTYPE 00_{16} is a bit-encoded value that indicates support for INFOTYPEs from 01_{16} to 20_{16}. INFOTYPE 20_{16} indicates support for INFOTYPEs 21_{16} through 40_{16}, etc. This is the same concept as used for PID support in Services 01_{16} and 02_{16} as specified in SAE J1979-DA.

The external test equipment shall maintain a list of ECUs which support the INFOTYPEs not equal to 00_{16} in order to justify whether it expects a response message from this ECU or not. For request messages with INFOTYPEs not equal to 00_{16}, the positive response messages may not be sent by the ECU(s) within the $P2_{max}$ timing window as specified in 6.2.2.

If INFOTYPE 02_{16} (VIN) is indicated as supported, the ECU shall respond within $P2_{max}$ timing even if the VIN is missing or incomplete. For example, a development ECU may respond with FF_{16} characters for VIN because the VIN has not been programmed.

7.9.2 Message data bytes

7.9.2.1 Request vehicle information request message definition (read-supported INFOTYPE)

Table 90 — Request vehicle information request message (read-supported INFOTYPE)

Data Byte	**Parameter Name**	**Cvt**	**Byte Value**	**Mnemonic**
#1	Request vehicle information request SID	M	09_{16}	SIDRQ
#2	INFOTYPE (see SAE J1979-DA)	M	XX_{16}	INFTYP

7.9.2.2 Request vehicle information response message definition (report supported INFOTYPE)

Table 91 — Request vehicle information response message (report supported INFOTYPE)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request vehicle information response SID	M	49_{16}	SIDPR
#2	INFOTYPE	M	XX_{16}	INFTYP_
#3	MessageCount	M	XX_{16}	MC
	data record of INFOTYPE = [			DATAREC_
#4	Data A: supported INFOTYPEs,	M	XX_{16}	DATA_A
#5	Data B: supported INFOTYPEs,	M	XX_{16}	DATA_B
#6	Data C: supported INFOTYPEs,	M	XX_{16}	DATA_C
#7	Data D: supported INFOTYPEs]	M	XX_{16}	DATA_D

7.9.2.3 Request vehicle information request message definition (read INFOTYPE values)

Table 92 — Request vehicle information request message (read INFOTYPE values)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request vehicle information request SID	M	09_{16}	SIDRQ
#2	INFOTYPE	M	XX_{16}	INFTYP_

7.9.2.4 Request vehicle information response message definition (report INFOTYPE values)

Table 93 — Request vehicle information response message (report INFOTYPE values)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request vehicle information response SID	M	49_{16}	SIDPR
#2	INFOTYPE	M	XX_{16}	INFTYP_
#3	MessageCount	M	XX_{16}	MC_
#4	data record of INFOTYPE = [Data A,	M/C[a]	XX_{16}	DATA_A
#5	Data B,	M/C	XX_{16}	DATA_B
#6	Data C,	M/C	XX_{16}	DATA_C
#7	Data D]	M/C	XX_{16}	DATA_D

[a] C = Conditional. Data A to D is only present if the requested INFOTYPE equals an even number.

7.9.3 Parameter definition

7.9.3.1 Vehicle information types supported

Refer to SAE J1979-DA.

7.9.3.2 Vehicle information types and data byte descriptions

Refer to SAE J1979-DA.

7.9.3.3 MessageCount description

The MessageCount parameter has two (2) definitions depending on the INFOTYPE parameter value:

— INFOTYPE parameter values 01_{16}, 03_{16}, 05_{16}, 07_{16}, 09_{16}, $0C_{16}$: In this case, the MessageCount parameter includes a value which represents the number of response messages to be sent by the server (ECU) to report the Data A to D referenced by the corresponding INFOTYPE parameter value. The MessageCount parameter value is a "static value".

— INFOTYPE parameter values 02_{16}, 04_{16}, 06_{16}, 08_{16}, $0A_{16}$, $0B_{16}$, $0D_{16}$: In this case, the MessageCount parameter includes a value which represents a dynamic counter starting with the value of 1 and incremented by 1 in the following response messages (assuming error-free transmission of the response message). The MessageCount parameter value is a "dynamic incremented value" (increments of 1). The last response message shall include an incremented MessageCount value which matches the reported MessageCount parameter value previously reported by the server (ECU) with the odd INFOTYPE (even INFOTYPE – 1).

Refer to SAE J1979-DA.

7.9.4 Message example

The tables below show how the "request vehicle information" service shall be implemented.

7.9.4.1 Step #1: Request vehicle information (request supported INFOTYPE) from vehicle

The external test equipment requests all supported INFOTYPEs from the vehicle. Refer to the example of Service 01_{16} for guidance on requesting supported PIDs (the same concept is used for supported INFOTYPEs). As a result of the supported INFOTYPE request, the external test equipment creates an internal list of supported INFOTYPEs for each ECU: ECU#1 (ECM) supports the following INFOTYPEs: 01_{16}, 02_{16}, 03_{16}, 04_{16}, 05_{16}, 06_{16}, 07_{16}, and 08_{16}. Since there is only one ECU which meets emission-related legislative requirements, no response messages from another ECU will occur.

7.9.4.2 Step #2: Request INFOTYPEs from vehicle

Now the external test equipment requests the following INFOTYPE:

— INFOTYPE 01_{16}: MC_VIN = 5 response messages; supported by ECU#1.

Table 94 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: MessageCount VIN	01_{16}	INFTYP

Table 95 — 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**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: MessageCount VIN	01_{16}	INFTYP
#3	MessageCount VIN = 5 response messages	05_{16}	MC_VIN

Now the external test equipment requests the following INFOTYPE:

— INFOTYPE 02_{16}: VIN = [1G1JC5444R7252367] supported by ECU#1.

Table 96 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: VIN	02_{16}	INFTYP

Table 97 — Request vehicle information response message (1)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: VIN	02_{16}	INFTYP
#3	MessageCount VIN = 1st response message	01_{16}	MC_VIN
#4	Data A: Filler Byte byte	00_{16}	DATA_A
#5	Data B: Filler Byte byte	00_{16}	DATA_B
#6	Data C: Filler Byte byte	00_{16}	DATA_C
#7	Data D: '1'	31_{16}	DATA_D

Table 98 — Request vehicle information response message (2)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: VIN	02_{16}	INFTYP
#3	MessageCount VIN = 2nd response message	02_{16}	MC_VIN
#4	Data A: 'G'	47_{16}	DATA_A
#5	Data B: '1'	31_{16}	DATA_B
#6	Data C: 'J'	$4A_{16}$	DATA_C
#7	Data D: 'C'	43_{16}	DATA_D

Table 99 — Request vehicle information response message (3)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: VIN	02_{16}	INFTYP
#3	MessageCount VIN = 3rd response message	03_{16}	MC_VIN
#4	Data A: '5'	35_{16}	DATA_A
#5	Data B: '4'	34_{16}	DATA_B
#6	Data C: '4'	34_{16}	DATA_C
#7	Data D: '4'	34_{16}	DATA_D

Table 100 — Request vehicle information response message (4)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: VIN	02_{16}	INFTYP
#3	MessageCount VIN = 4th response message	04_{16}	MC_VIN
#4	Data A: 'R'	52_{16}	DATA_A
#5	Data B: '7'	37_{16}	DATA_B
#6	Data C: '2'	32_{16}	DATA_C
#7	Data D: '5'	35_{16}	DATA_D

Table 101 — Request vehicle information response message (5)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: VIN	02_{16}	INFTYP
#3	MessageCount VIN = 5th response message	05_{16}	MC_VIN
#4	Data A: '2'	32_{16}	DATA_A
#5	Data B: '3'	33_{16}	DATA_B
#6	Data C: '6'	36_{16}	DATA_C
#7	Data D: '7'	37_{16}	DATA_D

Now the external test equipment requests the following INFOTYPE:

— INFOTYPE 03_{16}: MessageCount Calibration ID = 08_{16}; supported by ECU#1.

Table 102 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: MessageCount Calibration ID	03_{16}	INFTYP

Table 103 — 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**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: MessageCount Calibration ID	03_{16}	INFTYP
#3	MessageCount Calibration ID = 8 response messages	08_{16}	MC_CALID

Now the external test equipment requests the following INFOTYPE:

— INFOTYPE 04_{16}: CALID#1 = [JMB*36761500]; supported by ECU#1;

— INFOTYPE 04_{16}: CALID#2 = [JMB*47872611]; supported by ECU#1.

Table 104 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: Calibration ID	04_{16}	INFTYP

Table 105 — Request vehicle information response message (1)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Calibration ID	04_{16}	INFTYP
#3	MessageCount Calibration ID#1 = 1st response message	01_{16}	MC_CALID
#4	Data A: ‘J’	$4A_{16}$	DATA_A
#5	Data B: ‘M’	$4D_{16}$	DATA_B
#6	Data C: ‘B’	42_{16}	DATA_C
#7	Data D: ‘*’	$2A_{16}$	DATA_D

Table 106 — Request vehicle information response message (2)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Calibration ID	04_{16}	INFTYP
#3	MessageCount Calibration ID#1 = 2nd response message	02_{16}	MC_CALID
#4	Data A: '3'	33_{16}	DATA_A
#5	Data B: '6'	36_{16}	DATA_B
#6	Data C: '7'	37_{16}	DATA_C
#7	Data D: '6'	36_{16}	DATA_D

Table 107 — Request vehicle information response message (3)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Calibration ID	04_{16}	INFTYP
#3	MessageCount Calibration ID#1 = 3rd response message	03_{16}	MC_CALID
#4	Data A: '1'	31_{16}	DATA_A
#5	Data B: '5'	35_{16}	DATA_B
#6	Data C: '0'	30_{16}	DATA_C
#7	Data D: '0'	30_{16}	DATA_D

Table 108 — Request vehicle information response message (4)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Calibration ID	04_{16}	INFTYP
#3	MessageCount Calibration ID#1 = 4th response message	04_{16}	MC_CALID
#4	Data A: Filler Byte byte	00_{16}	DATA_A
#5	Data B: Filler Byte byte	00_{16}	DATA_B
#6	Data C: Filler Byte byte	00_{16}	DATA_C
#7	Data D: Filler Byte byte	00_{16}	DATA_D

Table 109 — Request vehicle information response message (5)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Calibration ID	04_{16}	INFTYP
#3	MessageCount Calibration ID#2 = 5th response message	05_{16}	MC_CALID
#4	Data A: 'J'	$4A_{16}$	DATA_A
#5	Data B: 'M'	$4D_{16}$	DATA_B
#6	Data C: 'B'	42_{16}	DATA_C
#7	Data D: '*'	$2A_{16}$	DATA_D

Table 110 — Request vehicle information response message (6)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Calibration ID	04_{16}	INFTYP
#3	MessageCount Calibration ID#2 = 6th response message	06_{16}	MC_CALID
#4	Data A: '4'	34_{16}	DATA_A
#5	Data B: '7'	37_{16}	DATA_B
#6	Data C: '8'	38_{16}	DATA_C
#7	Data D: '7'	37_{16}	DATA_D

Table 111 — Request vehicle information response message (7)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Calibration ID	04_{16}	INFTYP
#3	MessageCount Calibration ID#2 = 7th response message	07_{16}	MC_CALID
#4	Data A: '2'	32_{16}	DATA_A
#5	Data B: '6'	36_{16}	DATA_B
#6	Data C: '1'	31_{16}	DATA_C
#7	Data D: '1'	31_{16}	DATA_D

Table 112 — Request vehicle information response message (8)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Calibration ID	04_{16}	INFTYP
#3	MessageCount Calibration ID#2 = 8th response message	08_{16}	MC_CALID
#4	Data A: Filler Byte byte	00_{16}	DATA_A
#5	Data B: Filler Byte byte	00_{16}	DATA_B
#6	Data C: Filler Byte byte	00_{16}	DATA_C
#7	Data D: Filler Byte byte	00_{16}	DATA_D

Now the external test equipment requests the following INFOTYPE:

— INFOTYPE 05_{16}: MessageCount Calibration Verification Number = 02_{16}; supported by ECU#1.

Table 113 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: MessageCount Calibration Verification Number	05_{16}	INFTYP

Table 114 — 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**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: MessageCount Calibration Verification Number	05_{16}	INFTYP
#3	MessageCount Calibration Verification Number = 2 response messages	02_{16}	MC_CVN

Now the external test equipment requests the following INFOTYPE:

— INFOTYPE 06_{16}: CVN#1 = [17 91 BC 82]; supported by ECU#1;

— INFOTYPE 06_{16}: CVN#2 = [16 E0 62 BE]; supported by ECU#1.

Table 115 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: Calibration Verification Number	06_{16}	INFTYP

Table 116 — Request vehicle information response message (1)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Calibration Verification Number	06_{16}	INFTYP
#3	MessageCount Calibration Verification Number = 1st response message	01_{16}	MC_CVN
#4	Data A: 17	17_{16}	DATA_A
#5	Data B: 91	91_{16}	DATA_B
#6	Data C: BC	BC_{16}	DATA_C
#7	Data D: 82	82_{16}	DATA_D

Depending on which protocol the vehicle supports, the following situations might occur.

If the vehicle supports ISO 9141-2, the external test equipment might need to repeat the request message multiple times before the ECU(s) send a response message.

If the vehicle supports SAE J1850, the external test equipment might need to repeat the request message before the ECU(s) send a response message.

If the vehicle supports ISO 14230-4, the ECU(s) may send a negative response message with NRC 22_{16} - conditionsNotCorrect if, for example, the engine is running. After the vehicle conditions have been adjusted to meet this service request, the external test equipment shall repeat the request message and the ECU(s) shall send a positive response message.

Table 117 — Request vehicle information response message (2)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Calibration Verification Number	06_{16}	INFTYP
#3	MessageCount Calibration Verification Number = 2nd response message	02_{16}	MC_CVN
#4	Data A: 16_{16}	16_{16}	DATA_A
#5	Data B: $E0_{16}$	$E0_{16}$	DATA_B
#6	Data C: 62_{16}	62_{16}	DATA_C
#7	Data D: BE_{16}	BE_{16}	DATA_D

Now the external test equipment requests the following INFOTYPE:

— INFOTYPE 07_{16}: MessageCount In-use Performance Tracking = 08_{16}; supported by ECU#1.

Table 118 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: MessageCount In-use Performance Tracking	07_{16}	INFTYP

Table 119 — 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**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: MessageCount In-use Performance Tracking	07_{16}	INFTYP
#3	MessageCount In-use Performance Tracking = 8 response messages	08_{16}	MC_IPT

Now the external test equipment requests the following INFOTYPE:

— INFOTYPE 08_{16}: MC_IPT = 8 response messages; supported by ECU#1.

Table 120 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: In-use Performance Tracking	08_{16}	INFTYP

Table 121 — Request vehicle information response message (1)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: In-use Performance Tracking	08_{16}	INFTYP
#3	MessageCount In-use Performance Tracking = 1st response message	01_{16}	MC_IPT
#4	OBDCOND_A: 1024 counts	04_{16}	OBDCOND_A
#5	OBDCOND_B: 1024 counts	00_{16}	OBDCOND_B
#6	IGNCNTR_A: 3337 counts	$0D_{16}$	IGNCNTR_A
#7	IGNCNTR_B: 3337 counts	09_{16}	IGNCNTR_B

Table 122 — Request vehicle information response message (2)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: In-use Performance Tracking	08_{16}	INFTYP
#3	MessageCount In-use Performance Tracking = 2nd response message	02_{16}	MC_IPT
#4	CATCOMP1_A: 824 counts	03_{16}	CATCOMP1_A
#5	CATCOMP1_B: 824 counts	38_{16}	CATCOMP1_B
#6	CATCOND1_A: 945 counts	03_{16}	CATCOND1_A
#7	CATCOND1_B: 945 counts	$B1_{16}$	CATCOND1_B

Table 123 — Request vehicle information response message (3)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: In-use Performance Tracking	08_{16}	INFTYP
#3	MessageCount In-use Performance Tracking = 3rd response message	03_{16}	MC_IPT
#4	CATCOMP2_A: 711 counts	02_{16}	CATCOMP2_A
#5	CATCOMP2_B: 711 counts	$C7_{16}$	CATCOMP2_B
#6	CATCOND2_A: 945 counts	03_{16}	CATCOND2_A
#7	CATCOND2_B: 945 counts	$B1_{16}$	CATCOND2_B

Table 124 — Request vehicle information response message (4)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: In-use Performance Tracking	08_{16}	INFTYP
#3	MessageCount In-use Performance Tracking = 4th response message	04_{16}	MC_IPT
#4	O2SCOMP1_A: 737 counts	02_{16}	O2SCOMP1_A
#5	O2SCOMP1_B: 737 counts	$E1_{16}$	O2SCOMP1_B
#6	O2SCOND1_A: 924 counts	03_{16}	O2SCOND1_A
#7	O2SCOND1_B: 924 counts	$9C_{16}$	O2SCOND1_B

Table 125 — Request vehicle information response message (5)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: In-use Performance Tracking	08_{16}	INFTYP
#3	MessageCount In-use Performance Tracking = 5th response message	05_{16}	MC_IPT
#4	O2SCOMP2_A: 724 counts	02_{16}	O2SCOMP2_A
#5	O2SCOMP2_B: 724 counts	$D4_{16}$	O2SCOMP2_B
#6	O2SCOND2_A: 833 counts	03_{16}	O2SCOND2_A
#7	O2SCOND2_B: 833 counts	41_{16}	O2SCOND2_B

Table 126 — Request vehicle information response message (6)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: In-use Performance Tracking	08_{16}	INFTYP
#3	MessageCount In-use Performance Tracking = 6th response message	06_{16}	MC_IPT
#4	EGRCOMP_A: 997 counts	03_{16}	EGRCOMP_A
#5	EGRCOMP_B: 997 counts	$E5_{16}$	EGRCOMP_B
#6	EGRCOND_A: 1010 counts	03_{16}	EGRCOND_A
#7	EGRCOND_B: 1010 counts	$F2_{16}$	EGRCOND_B

Table 127 — Request vehicle information response message (7)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: In-use Performance Tracking	08_{16}	INFTYP
#3	MessageCount In-use Performance Tracking = 7th response message	07_{16}	MC_IPT
#4	AIRCOMP_A: 937 counts	03_{16}	AIRCOMP_A
#5	AIRCOMP_B: 937 counts	$A9_{16}$	AIRCOMP_B
#6	AIRCOND_A: 973 counts	03_{16}	AIRCOND_A
#7	AIRCOND_B: 973 counts	CD_{16}	AIRCOND_B

Table 128 — Request vehicle information response message (8)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: In-use Performance Tracking	08_{16}	INFTYP
#3	MessageCount In-use Performance Tracking = 8th response message	08_{16}	MC_IPT
#4	EVAPCOMP_A: 68 counts	00_{16}	EVAPCOMP_A
#5	EVAPCOMP_B: 68 counts	44_{16}	EVAPCOMP_B
#6	EVAPCOND_A: 97 counts	00_{16}	EVAPCOND_A
#7	EVAPCOND_B: 97 counts	61_{16}	EVAPCOND_B

Now the external test equipment requests the following InfoType:

— InfoType 09_{16}: MessageCount ECUName = $0A_{16}$; supported by ECU#1.

Table 129 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: MessageCount ECUName	09_{16}	INFTYP

Table 130 — 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**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: MessageCount ECUName	09_{16}	INFTYP
#3	MessageCount ECUName = 5 response messages	05_{16}	MC_ECUNM

Now the external test equipment requests the following InfoType

— InfoType $0A_{16}$: ECUName = [ECM –EngineControl] supported by ECU#1.

Table 131 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: ECUName	$0A_{16}$	INFTYP

Table 132 — Request vehicle information response message (1)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: ECUName	$0A_{16}$	INFTYP
#3	MessageCount ECUName = 1st response message	01_{16}	MC_ECUNM
#4	DATA A: 'E'	45_{16}	DATA A
#5	DATA B. 'C'	43_{16}	DATA B
#6	DATA C: 'M'	$4D_{16}$	DATA C
#7	DATA D: 00_{16}	00_{16}	DATA D

Table 133 — Request vehicle information response message (2)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: ECUName	$0A_{16}$	INFTYP
#3	MessageCount In-use ECUName = 2nd response message	02_{16}	MC_ECUNM
#4	DATA A: '-'	$2D_{16}$	DATA A
#5	DATA B. 'E'	45_{16}	DATA B
#6	DATA C: 'n'	$6E_{16}$	DATA C
#7	DATA D: 'g'	67_{16}	DATA D

Table 134 — Request vehicle information response message (3)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: ECUName	$0A_{16}$	INFTYP
#3	MessageCount In-use ECUName = 3rd response message	03_{16}	MC_ECUNM
#4	DATA A: 'i'	69_{16}	DATA A
#5	DATA B. 'n'	$6E_{16}$	DATA B
#6	DATA C: 'e'	65_{16}	DATA C
#7	DATA D: 'C'	43_{16}	DATA D

Table 135 — Request vehicle information response message (4)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	InfoType: ECUName	$0A_{16}$	INFTYP

Table 135 *(continued)*

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#3	MessageCount ECUName = 4th response message	04_{16}	MC_ECUNM
#4	DATA A: 'o'	$6F_{16}$	DATA A
#5	DATA B. 'n'	$6E_{16}$	DATA B
#6	DATA C: 't'	74_{16}	DATA C
#7	DATA D: 'r'	72_{16}	DATA D

Table 136 — Request vehicle information response message (5)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: ECUName	$0A_{16}$	INFTYP
#3	MessageCount ECUName = 5th response message	05_{16}	MC_ECUNM
#4	DATA A: 'o'	$6F_{16}$	DATA A
#5	DATA B. 'l'	$6C_{16}$	DATA B
#6	DATA C: 00_{16}	00_{16}	DATA C
#7	DATA D: 00_{16}	00_{16}	DATA D

Now the external test equipment requests the following InfoType:

— InfoType $0C_{16}$: MessageCount ESN = $0D_{16}$; supported by ECU#1.

Table 137 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: MessageCount ESN	$0C_{16}$	INFTYP

Table 138 — 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**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: MessageCount ESN	$0C_{16}$	INFTYP
#3	MessageCount ESN = 5 response messages	05_{16}	MC_ESN

Now the external test equipment requests the following InfoType:

— InfoType $0D_{16}$: Engine Serial Number = [BRAND 3217486] supported by ECU#1.

Table 139 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: ESN	$0D_{16}$	INFTYP

Table 140 — Request vehicle information response message (1)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Engine Serial Number	$0D_{16}$	INFTYP
#3	MessageCount ESN = 1st response message	01_{16}	MC_ESN
#4	DATA A: 00_{16}	00_{16}	DATA A
#5	DATA B. 00_{16}	00_{16}	DATA B
#6	DATA C: 00_{16}	00_{16}	DATA C
#7	DATA D: 00_{16}	00_{16}	DATA D

Table 141 — Request vehicle information response message (2)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Engine Serial Number	$0D_{16}$	INFTYP
#3	MessageCount In-use ESN = 2nd response message	02_{16}	MC_ESN
#4	DATA A: 00_{16}	00_{16}	DATA A
#5	DATA B. 00_{16}	00_{16}	DATA B
#6	DATA C: 00_{16}	00_{16}	DATA C
#7	DATA D: ‘B’	42_{16}	DATA D

Table 142 — Request vehicle information response message (3)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Engine Serial Number	$0D_{16}$	INFTYP
#3	MessageCount In-use ESN = 3rd response message	03_{16}	MC_ESN
#4	DATA A: ‘R’	52_{16}	DATA A
#5	DATA B. ‘A’	41_{16}	DATA B
#6	DATA C: ‘N’	$4E_{16}$	DATA C
#7	DATA D: ‘D’	44_{16}	DATA D

Table 143 — Request vehicle information response message (4)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Engine Serial Number	$0D_{16}$	INFTYP
#3	MessageCount ESN = 4th response message	04_{16}	MC_ESN
#4	DATA A: ‘ ’	20_{16}	DATA A
#5	DATA B. ‘3‘	33_{16}	DATA B
#6	DATA C: ‘2	32_{16}	DATA C
#7	DATA D: ‘1	31_{16}	DATA D

Table 144 — Request vehicle information response message (5)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Engine Serial Number	$0D_{16}$	INFTYP
#3	MessageCount ESN = 5th response message	05_{16}	MC_ESN
#4	DATA A: ‘7’	37_{16}	DATA A
#5	DATA B. ‘4’	34_{16}	DATA B
#6	DATA C: ‘8’	38_{16}	DATA C
#7	DATA D: ‘6’	36_{16}	DATA D

Now the external test equipment requests the following InfoType:

— InfoType $0E_{16}$: Exhaust Regulation Or Type Approval Number = [DOC-CR-934567] supported by ECU#1.

Table 145 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: MessageCount EROTAN	$0E_{16}$	INFTYP

Table 146 — 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**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: MessageCount EROTAN	$0E_{16}$	INFTYP
#3	MessageCount EROTAN = 5 response messages	05_{16}	MC_EROTAN

Now the external test equipment requests the following InfoType:

— InfoType $0F_{16}$: MC_EROTAN = 5 response messages; supported by ECU#1.

Table 147 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: EROTAN	$0F_{16}$	INFTYP

Table 148 — Request vehicle information response message (1)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: EROTAN	$0F_{16}$	INFTYP
#3	MessageCount EROTAN = 1st response message	01_{16}	MC_EROTAN
#4	DATA A: 00_{16}	00_{16}	DATA A
#5	DATA B. 00_{16}	00_{16}	DATA B
#6	DATA C: 00_{16}	00_{16}	DATA C
#7	DATA D: 00_{16}	00_{16}	DATA D

Table 149 — Request vehicle information response message (2)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: EROTAN	$0F_{16}$	INFTYP
#3	MessageCount EROTAN = 2nd response message	02_{16}	MC_EROTAN
#4	DATA A: 00_{16}	00_{16}	DATA A
#5	DATA B. 00_{16}	00_{16}	DATA B
#6	DATA C: 00_{16}	00_{16}	DATA C
#7	DATA D: ‘D’	44_{16}	DATA D

Table 150 — Request vehicle information response message (3)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: EROTAN	$0F_{16}$	INFTYP
#3	MessageCount EROTAN = 3rd response message	03_{16}	MC_EROTAN
#4	DATA A: ‘O’	$4F_{16}$	DATA A

Table 150 *(continued)*

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#5	DATA B. 'C'	43_{16}	DATA B
#6	DATA C: '-'	$2D_{16}$	DATA C
#7	DATA D: 'C'	43_{16}	DATA D

Table 151 — Request vehicle information response message (4)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: In-use Performance Tracking	$0F_{16}$	INFTYP
#3	MessageCount EROTAN = 4th response message	04_{16}	MC_EROTAN
#4	DATA A: 'R'	52_{16}	DATA A
#5	DATA B. '-'	$2D_{16}$	DATA B
#6	DATA C: '9'	39_{16}	DATA C
#7	DATA D: '3'	33_{16}	DATA D

Table 152 — Request vehicle information response message (5)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: EROTAN	$0F_{16}$	INFTYP
#3	MessageCount EROTAN = 5th response message	05_{16}	MC_EROTAN
#4	DATA A: '4'	34_{16}	DATA A
#5	DATA B. '5'	35_{16}	DATA B
#6	DATA C: '6'	36_{16}	DATA C
#7	DATA D: '7'	37_{16}	DATA D

8 Diagnostic service definition for ISO 15765-4

8.1 Service 01_{16} — Request current powertrain diagnostic data

8.1.1 Functional description

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 parameter identification (PID) value that indicates to the on-board system the specific information requested. PID specifications, scaling information, and display formats are included in SAE J1979-DA.

The ECU(s) shall respond to this message by transmitting the requested data value last determined by the system. All data values returned for sensor readings shall be actual readings, not default or substitute values used by the system because of a fault with that sensor.

Not all PIDs are applicable or supported by all systems. PID 00_{16} is a bit-encoded value that indicates which PIDs are supported for each ECU. PID 00_{16} indicates support for PIDs from 01_{16} to 20_{16}. PID 20_{16} indicates support for PIDs 21_{16} through 40_{16}, etc. This is the same concept for PIDs/OBD Monitor IDs/TIDs/INFOTYPEs support in Services 01_{16}, 02_{16}, 06_{16}, 08_{16}, 09_{16}. PID 00_{16} is required for those ECUs that respond to a corresponding Service 01_{16} request message as specified in SAE J1979-DA.

IMPORTANT — All emissions-related OBD ECUs which at least support one of the services defined in this part of ISO 15031, shall support Service 01_{16} and PID 00_{16}. Service 01_{16} with PID 00_{16} is defined as the universal "initialization/keep alive/ping" message for all emissions-related OBD ECUs.

The request message may contain up to six (6) PIDs. External test equipment is not allowed to request a combination of PIDs supported and PIDs which report data values. The ECU shall support requests for up to six (6) PIDs. The request message may contain the same PID multiple times. The ECU shall treat each PID as a separate parameter and respond with data for each PID (data returned may be different for the same PID) as often as requested.

The order of the PIDs in the response message is not required to match the order in the request message.

8.1.2 Message data bytes

8.1.2.1 Request current powertrain diagnostic data request message definition (read-supported PIDs)

Table 153 — Request current powertrain diagnostic data request message (read-supported PIDs)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request current powertrain diagnostic data request SID	M	01_{16}	SIDRQ
#2	PID#1 (PIDs supported: see SAE J1979-DA)	M	XX_{16}	PID
#3	PID#2 (PIDs supported: see SAE J1979-DA)	U[a]	XX_{16}	PID
#4	PID#3 (PIDs supported: see SAE J1979-DA)	U	XX_{16}	PID
#5	PID#4 (PIDs supported: see SAE J1979-DA)	U	XX_{16}	PID
#6	PID#5 (PIDs supported: see SAE J1979-DA)	U	XX_{16}	PID
#7	PID#6 (PIDs supported: see SAE J1979-DA)	U	XX_{16}	PID

[a] U = User Optional. PID may be included to avoid multiple PID supported request messages.

To request PIDs supported range from $C1_{16}$ to FF_{16}, another request message with PID#1 = $C0_{16}$ and PID#2 = $E0_{16}$ shall be sent to the vehicle.

8.1.2.2 Request current powertrain diagnostic data response message definition (report supported PIDs)

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 01_{16} - 20_{16}). The ECU shall not respond to unsupported PID ranges unless subsequent ranges have a supported PID(s).

Table 154 — Request current powertrain diagnostic data response message (report supported PIDs)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request current powertrain diagnostic data response SID	M	41_{16}	SIDPR
	data record of supported PIDs = [			PIDREC_
#2	1st supported PID	M	XX_{16}	PID
#3	Data A: supported PIDs,	M	XX_{16}	DATA_A
#4	Data B: supported PIDs,	M	XX_{16}	DATA_B
#5	Data C: supported PIDs,	M	XX_{16}	DATA_C
#6	Data D: supported PIDs]	M	XX_{16}	DATA_D
:	:	:	:	:
	data record of supported PIDs = [			PIDREC_
#n-4	mth supported PID	C1[a]	XX_{16}	PID
#n-3	Data A: supported PIDs,	C2[b]	XX_{16}	DATA_A
#n-2	Data B: supported PIDs,	C2	XX_{16}	DATA_B
#n-1	Data C: supported PIDs,	C2	XX_{16}	DATA_C
#n	Data D: supported PIDs]	C2	XX_{16}	DATA_D

[a] C1 = Conditional. PID value shall be the same value as included in the request message if supported by the ECU.

[b] C2 = Conditional. Value indicates PIDs supported; range of supported PIDs depends on selected PID value (see C1).

The response message shall only include the PID(s) and Data A to D which are supported by the ECU. If the request message includes a PID value(s) which is (are) not supported by the ECU, those shall not be included in the response message.

8.1.2.3 Request current powertrain diagnostic data request message definition (read PID values)

Table 155 — Request current powertrain diagnostic data request message

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request current powertrain diagnostic data request SID	M	01_{16}	SIDRQ
#2	PID#1 (see SAE J1979-DA)	M	XX_{16}	PID
#3	PID#2 (see SAE J1979-DA)	U[a]	XX_{16}	PID
#4	PID#3 (see SAE J1979-DA)	U	XX_{16}	PID
#5	PID#4 (see SAE J1979-DA)	U	XX_{16}	PID
#6	PID#5 (see SAE J1979-DAB)	U	XX_{16}	PID
#7	PID#6 (see SAE J1979-DAB)	U	XX_{16}	PID

[a] U = User Optional. The parameter may be either present or not.

8.1.2.4 Request current powertrain diagnostic data response message definition (report PID values)

Table 156 — Request current powertrain diagnostic data response message

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request current powertrain diagnostic data response SID	M	41_{16}	SIDPR
	data record of 1st supported PID = [			PIDREC_
#2	PID#1	M	XX_{16}	PID
#3	data #1.1,	M	XX_{16}	DATA_1.1
:	:	:	XX_{16}	:
#j+1	data #1.j]	C1[a]	XX_{16}	DATA_1.j
:	:	:	:	:
	data record of mth supported PID = [			PIDREC_
:	PID#m	C2[b]	XX_{16}	PID
:	data #m.1,	C2	XX_{16}	DATA_m.1
:	:	:	:	:
#n	data #m.k]	C3[c]	XX_{16}	DATA_1.k

[a] C1 = Conditional. Data depends on selected PID value.

[b] C2 = Conditional. Parameter is only present if supported by the ECU.

[c] C3 = Conditional. Parameters and values for data depend on selected PID number and are only included if PID is supported by the ECU.

Not all PIDs which are included in the request message may be supported by all emission-related ECUs, which shall comply with this part of ISO 15031. Therefore, each vehicle ECU, which supports at least one (1) PID, shall send a response message including the PID(s) with data.

8.1.3 Parameter definition

8.1.3.1 PIDs supported

SAE J1979-DA specifies the interpretation of the data record of supported PIDs.

8.1.3.2 PID and data byte descriptions

SAE J1979-DA specifies standardized emission-related parameters.

8.1.4 Message example

Tables 133 to 135 show how the “request current powertrain diagnostic data” service shall be implemented.

8.1.4.1 Step #1: Request supported PIDs from vehicle

The external test equipment requests supported PIDs (00_{16}, 20_{16}, 40_{16}, 60_{16}, 80_{16}, $A0_{16}$) from the vehicle. Refer to SAE J1979-DA to interpret the data bytes in the response messages.

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 01_{16} - 20_{16}). The ECU shall not respond to unsupported PID ranges unless subsequent ranges have a supported PID(s).

Table 157 — 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**	**Mnemonic**
#1	Request current powertrain diagnostic data request SID	01_{16}	SIDRQ
#2	PID used to determine PID support for PIDs 01_{16} - 20_{16}	00_{16}	PID
#3	PID used to determine PID support for PIDs 21_{16} – 40_{16}	20_{16}	PID
#4	PID used to determine PID support for PIDs 41_{16} – 60_{16}	40_{16}	PID
#5	PID used to determine PID support for PIDs 61_{16} – 80_{16}	60_{16}	PID
#6	PID used to determine PID support for PIDs 81_{16} – $A0_{16}$	80_{16}	PID
#7	PID used to determine PID support for PIDs $A1_{16}$ – $C0_{16}$	$A0_{16}$	PID

Table 158 — 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**	**Mnemonic**
#1	Request current powertrain diagnostic data response SID	41_{16}	SIDPR
#2	PID requested	00_{16}	PID
#3	Data byte A, representing support for PIDs 01_{16}, 03_{16} - 08_{16}	$10111111_2 = BF_{16}$	DATA_A
#4	Data byte B, representing support for PIDs 09_{16}, $0B_{16}$ - 10_{16}	$10111111_2 = BF_{16}$	DATA_B
#5	Data byte C, representing support for PIDs 11_{16}, 13_{16}, 15_{16}	$10101000_2 = A8_{16}$	DATA_C
#6	Data byte D, representing support for PIDs 19_{16}, $1C_{16}$, 20_{16}	$10010001_2 = 91_{16}$	DATA_D
#7	PID requested	20_{16}	PID
#8	Data byte A, representing support for PID 21_{16}	$10000000_2 = 80_{16}$	DATA_A
#9	Data byte B, representing no support for PIDs 29_{16} - 30_{16}	$00000000_2 = 00_{16}$	DATA_B
#10	Data byte C, representing no support for PIDs 31_{16} - 38_{16}	$00000000_2 = 00_{16}$	DATA_C
#11	Data byte D, representing no support for PIDs 39_{16} - 40_{16}	$00000000_2 = 00_{16}$	DATA_D

Table 159 — 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**	**Mnemonic**
#1	Request current powertrain diagnostic data response SID	41_{16}	SIDPR
#2	PID requested	00_{16}	PID
#3	Data byte A, representing support for PID 01_{16}	$10000000_2 = 80_{16}$	DATA_A
#4	Data byte B, representing support for PID $0D_{16}$	$00001000_2 = 08_{16}$	DATA_B
#5	Data byte C, representing no support for PIDs 11_{16} - 18_{16}	$00000000_2 = 00_{16}$	DATA_C
#6	Data byte D, representing no support for PIDs 19_{16} - 20_{16}	$00000000_2 = 00_{16}$	DATA_D

Now the external test equipment creates an internal list of supported PIDs for each ECU. ECU#1 (ECM) supports the following PIDs: 01_{16}, 03_{16} - 09_{16}, $0B_{16}$ - 11_{16}, 13_{16}, 15_{16}, 19_{16}, $1C_{16}$, 20_{16}, 21_{16}.

ECU#2 (TCM) supports the following PIDs: 01_{16} and $0D_{16}$.

8.1.4.2 Step #2: Request multiple PIDs from vehicle

Now 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 15_{16}: Bank 1 - Sensor 2, PID is supported by ECU#1;

— PID 01_{16}: Number of emission-related DTCs and MIL status, PID is supported by ECU#1 and #2;

— PID 05_{16}: Engine coolant temperature, PID is supported by ECU#1;

— PID 03_{16}: Fuel system 1 status, PID is supported by ECU#1;

— PID $0C_{16}$: Engine speed, PID is supported by ECU#1;

— PID $0D_{16}$: Vehicle speed, PID is supported by ECU#2.

Table 160 — 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**	**Mnemonic**
#1	Request current powertrain diagnostic data request SID	01_{16}	SIDRQ
#2	PID: Bank 1 - Sensor 2	15_{16}	PID(15)
#3	PID: Number of emission-related DTCs and MIL status	01_{16}	PID(01)
#4	PID: Engine coolant temperature	05_{16}	PID(05)
#5	PID: Fuel system 1 status	03_{16}	PID(03)
#6	PID: Engine speed	$0C_{16}$	PID(0C)
#7	PID: Vehicle speed	$0D_{16}$	PID(0D)

Table 161 — 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**	**Mnemonic**
#1	Request current powertrain diagnostic data response SID	41_{16}	SIDPR
#2	PID #1: Engine coolant temperature	05_{16}	PID(05)
#3	Data #1.1	$6E_{16}$	DATA(A)
#4	PID #2: Number of emission-related DTCs and MIL status	01_{16}	PID(01)
#5	Data #2.1: MIL: ON; Number of emission-related DTCs: 03	83_{16}	DATA(A)
#6	Data #2.2: Misfire -, Fuel system -, Comprehensive monitoring	07_{16}	DATA(B)
#7	Data #2.3: Catalyst -, Heated catalyst -, ..., monitoring supported	EF_{16}	DATA(C)
#8	Data #2.4: Catalyst -, Heated catalyst -, ..., monitoring test complete/not complete	63_{16}	DATA(D)
#9	PID #3: Bank 1 - Sensor 2	15_{16}	PID(15)
#10	Data #3.1: Bank 2 - Sensor 2: 0,8 Volt	$A0_{16}$	DATA(A)
#11	Data #3.2: Bank 2 - Sensor 2: 93,7 %	78_{16}	DATA(B)
#12	PID #4: Engine speed	$0C_{16}$	PID(0C)
#13	Data #4.1: 667 rpm	$0A_{16}$	DATA(A)

Table 161 *(continued)*

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (All PID values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#14	Data #4.2: 667 rpm	$6B_{16}$	DATA(B)
#15	PID #5: Fuel system 1 status	03_{16}	PID(03)
#16	Data #5.1: Closed loop - using oxygen sensor(s) as feedback for fuel control	02_{16}	DATA(A)
#17	Data #5.2:	00_{16}	DATA(B)

Table 162 — 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**	**Mnemonic**
#1	Request current powertrain diagnostic data response SID	41_{16}	SIDPR
#2	PID #1: Vehicle speed	$0D_{16}$	PID(0D)
#3	Data #1.1:	23_{16}	DATA(A)
#4	PID #2: Number of emission-related DTCs and MIL status	01_{16}	PID(01)
#5	Data #2.1: MIL: OFF; Number of emission-related DTCs: 01	01_{16}	DATA(A)
#6	Data #2.2: Comprehensive monitoring: supported, test complete	04_{16}	DATA(B)
#7	Data #2.3: Catalyst -, Heated catalyst -, ..., monitoring supported	00_{16}	DATA(C)
#8	Data #2.4: Catalyst -, Heated catalyst -, ..., monitoring test complete/not complete	00_{16}	DATA(D)

ECU #1 (ECM) reports MIL commanded on, three stored DTCs, all monitors as supported, catalyst, heated catalyst, oxygen sensor and oxygen sensor heater as not completed, and all other monitors as completed.

ECU #2 (TCM) reports MIL commanded off, one stored DTC, comprehensive components monitor as supported and complete, and all other monitors as not supported.

8.2 Service 02_{16} — Request powertrain freeze frame data

8.2.1 Functional description

The purpose of this service is to allow access to emission-related data values in a freeze frame. This allows expansion to meet manufacturer-specific requirements not necessarily related to the required freeze frame and not necessarily containing the same data values as the required freeze frame. The request message includes a parameter identification (PID) value that indicates to the on-board system the specific information requested. PID specifications, scaling information, and display formats for the freeze frame are included in SAE J1979-DA.

The ECU(s) shall respond to this message by transmitting the requested data value stored by the system. All data values returned for sensor readings shall be actual stored readings, not default or substitute values used by the system because of a fault with that sensor.

Service 02_{16} PID 02_{16} indicates the DTC that caused the freeze frame data to be stored. If freeze frame data are not stored in the ECU, the system shall report 0000_{16} as the DTC.

The frame number byte shall indicate 00_{16} for the freeze frame data. Manufacturers may optionally save additional freeze frames and use this service to obtain that data by specifying the freeze frame

number in the request message. If a manufacturer uses these additional freeze frames, they shall be stored under conditions specified by the manufacturer and contain data specified by the manufacturer.

Not all PIDs are applicable or supported by all systems. PID 00_{16} is a bit-encoded value that indicates for each ECU, for each frame, which PIDs are supported. Different freeze frames can support a different set of PIDs depending on the DTC that caused the frame to be stored. PID 00_{16} indicates support for PIDs from 01_{16} to 20_{16}. PID 20_{16} indicates support for PIDs 21_{16} through 40_{16}, etc. This is the same concept for PIDs/TIDs/INFOTYPEs support in Services 01_{16}, 02_{16}, 06_{16}, 08_{16}, 09_{16}. PID 00_{16} is required for those ECUs that respond to a corresponding Service 02_{16} request message as specified in SAE J1979-DA.

The order of the PIDs in the response message is not required to match the order in the request message.

External test equipment shall not request a combination of PIDs supported and PIDs which report data values. The ECU shall support requests for up to three (3) PIDs. The request message may contain the same PID multiple times. The ECU shall treat each PID as a separate parameter and respond with data for each PID as often as requested.

8.2.2 Message data bytes

8.2.2.1 Request powertrain freeze frame data request message definition (read-supported PIDs)

Table 163 — Request powertrain freeze frame data request message (read-supported PIDs)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request powertrain freeze frame data request SID	M	02_{16}	SIDRQ
#2	PID#1 (PIDs supported: SAE J1979-DA)	M	XX_{16}	PID
#3	frame #	M	XX_{16}	FRNO_
#4	PID#2 (PIDs supported: SAE J1979-DA)	U[a]	XX_{16}	PID
#5	frame #	U/C[b]	XX_{16}	FRNO_
#6	PID#3 (PIDs supported: SAE J1979-DA)	U	XX_{16}	PID
#7	frame #	U/C	XX_{16}	FRNO_

[a] U = User Optional. PID may be included to reduce multiple PID supported request messages.

[b] C = Conditional. Parameter is only included if the preceding PID# is included.

To request PIDs supported range from 61_{16} - FF_{16}, multiple request messages with PIDs = 60_{16}, 80_{16}, $A0_{16}$, $C0_{16}$, and $E0_{16}$ shall be sent to the vehicle.

8.2.2.2 Request powertrain freeze frame data response message definition (report supported PIDs)

The 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 01_{16} - 20_{16}). The ECU shall not respond to unsupported PID ranges unless subsequent ranges have a supported PID(s).

Table 164 — Request powertrain freeze frame data response message (report supported PIDs)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request powertrain freeze frame data response SID	M	42_{16}	SIDPR
#2	1st supported PID	M	00_{16}	PID

[a] C1 = Conditional. PID value shall be the same value as included in the request message if supported by the ECU.

[b] C2 = Conditional. Value indicates PIDs supported; range of supported PIDs depends on selected PID value (see C1).

Table 164 *(continued)*

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#3	frame #	M	XX_{16}	FRNO_
	data record of supported PIDs = [			DATAREC_
#4	Data A: supported PIDs,	M	XX_{16}	DATA_A
#5	Data B: supported PIDs,	M	XX_{16}	DATA_B
#6	Data C: supported PIDs,	M	XX_{16}	DATA_C
#7	Data D: supported PIDs]	M	XX_{16}	DATA_D
:	:	:	:	:
#n-5	m^{th} supported PID	C1[a]	XX_{16}	PID
#n-4	frame #	C1	XX_{16}	FRNO_
	data record of supported PIDs = [			DATAREC
#n-3	Data A: supported PIDs,	C2[b]	XX_{16}	DATA_A
#n-2	Data B: supported PIDs,	C2	XX_{16}	DATA_B
#n-1	Data C: supported PIDs,	C2	XX_{16}	DATA_C
#n	Data D: supported PIDs]	C2	XX_{16}	DATA_D

[a] C1 = Conditional. PID value shall be the same value as included in the request message if supported by the ECU.

[b] C2 = Conditional. Value indicates PIDs supported; range of supported PIDs depends on selected PID value (see C1).

The response message shall only include the PID(s) and Data A to D which are supported by the ECU. If the request message includes a PID value(s) which is (are) not supported by the ECU, those shall not be included in the response message.

8.2.2.3 Request powertrain freeze frame data request message definition (read freeze frame PID values)

Table 165 — Request powertrain freeze frame data request message (read freeze frame PID values)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request powertrain freeze frame data request SID	M	02_{16}	SIDRQ
#2	PID#1 (see SAE J1979-DA)	M	XX_{16}	PID
#3	frame #	M	XX_{16}	FRNO
#4	PID#2 (see SAE J1979-DA)	U[a]	XX_{16}	PID
#5	frame #	C1[b]	XX_{16}	FRNO
#6	PID#3 (see SAE J1979-DA)	U	XX_{16}	PID
#7	frame #	C1	XX_{16}	FRNO

[a] U = User Optional. The parameter may be either present or not.

[b] C1 = Conditional. Parameter is only present if the preceding PID# is present.

8.2.2.4 Request powertrain freeze frame data response message definition (report freeze frame PID values)

Table 166 — Request powertrain freeze frame data response message (report freeze frame PID values)

Data Byte	Parameter Name		Cvt	Byte Value	Mnemonic
#1	Request powertrain freeze frame data response SID		M	42_{16}	SIDPR
#2	1st supported PID		M	XX_{16}	PID
#3	frame #		M	XX_{16}	FRNO_
	data record of 1st supported PID = [				PIDREC_
#4		data #1.1,	M	XX_{16}	DATA_1.1
#5		data #1.2,	C1[a]	XX_{16}	DATA_1.2
:		:	:	XX_{16}	:
#j+3		data #1.j]	C1	XX_{16}	DATA_1.j
:	:		:	:	:
#n	mth supported PID		C2[b]	XX_{16}	PID_
#n+1	frame #		C2	XX_{16}	FRNO_
	data record of mth supported PID = [				PIDREC_
#n+2		data #m.1,	C4[c]	XX_{16}	DATA_m.1
#n+3		data #m.2,	C4[d]	XX_{16}	DATA_m.2
:		:	:	:	:
#n+k+1		data #m.k]	C4	XX_{16}	DATA_m.k

[a] C1 = Conditional. Data depends on selected PID.

[b] C2 = Conditional. Parameter shall be the same value as included in the request message and only present if supported.

[c] C3 = Conditional. Data #m.1 shall be included if preceding PID is supported.

[d] C4 = Conditional. Parameters and values for data depend on selected PID number.

8.2.3 Parameter definition

8.2.3.1 PIDs supported

SAE J1979-DA specifies the interpretation of the data record of supported PIDs.

8.2.3.2 PID and data byte descriptions

SAE J1979-DA specifies standardized emission-related parameters.

8.2.3.3 Frame number description

The frame number identifies the freeze frame, which includes emission-related data values in case an emission-related DTC is detected by the ECU.

8.2.4 Message example

The tables below show how the “request powertrain freeze frame data” service shall be implemented.

8.2.4.1 Step #1: Request supported powertrain freeze frame PIDs from vehicle

The external test equipment requests all supported powertrain freeze frame PIDs of freeze frame 00_{16} from the vehicle. Refer to the example of Service 01_{16} for guidance on requesting supported PIDs.

As a result of the supported PID request, the external test equipment creates an internal list of supported PIDs for each ECU. ECU#1 (ECM) supports the following PIDs: 02_{16} - 09_{16}, $0B_{16}$ - $0E_{16}$. ECU#2 (TCM) does not support any PIDs for this service.

8.2.4.2 Step #2: Request PID 02_{16} "DTC which caused freeze frame to be stored" from vehicle

Case #1: Freeze frame data are stored in ECU#1:

Now the external test equipment requests PID 02_{16} of freeze frame 00_{16} from the vehicle. Since ECU#2 (TCM) does not store a freeze frame data record, only 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 02_{16} "DTC that caused required freeze frame data storage" is set to the DTC P0130.

Table 167 — 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	02_{16}	SIDRQ
#2	PID: DTC that caused required freeze frame data storage	02_{16}	PID
#3	Frame #	00_{16}	FRNO

Table 168 — 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	42_{16}	SIDRQ
#2	PID: DTC that caused required freeze frame data storage	02_{16}	PID
#3	Frame #	00_{16}	FRNO
#4	DTC High Byte of P0130	01_{16}	DATA_A
#5	DTC Low Byte of P0130	30_{16}	DATA_B

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

Now the external test equipment requests the parameter value of PID $0C_{16}$ "Engine Speed", PID 05_{16} "Engine coolant temperature", and PID 04_{16} "Load", stored in the freeze frame.

Table 169 — 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	02_{16}	SIDRQ
#2	PID: Engine Speed	$0C_{16}$	PID
#3	Frame #	00_{16}	FRNO
#4	PID: Engine coolant temperature	05_{16}	PID
#5	Frame #	00_{16}	FRNO
#4	PID: Load	04_{16}	PID
#5	Frame #	00_{16}	FRNO

Table 170 — 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	42_{16}	SIDRQ
#2	PID: Engine Speed	$0C_{16}$	PID
#3	Frame #	00_{16}	FRNO
#4	High Byte: Engine Speed: 2080 rpm	20_{16}	DATA_A
#5	Low Byte: Engine Speed: 2080 rpm	80_{16}	DATA_B
#6	PID: Load	04_{16}	PID
#7	Frame #	00_{16}	FRNO
#8	Load: 50,2 %	80_{16}	DATA_A
#9	PID: Engine coolant temperature	05_{16}	PID
#10	Frame #	00_{16}	FRNO
#11	Engine coolant temperature: 0 °C	28_{16}	DATA_A

Case #2: No freeze frame data are stored in any ECU:

If no freeze frame data are stored, then the parameter value of PID 02_{16} "DTC that caused required freeze frame data storage" is set to 0000_{16}. If the external test equipment requests a PID excluding 00_{16}, 02_{16}, 20_{16}, 40_{16}, etc., the ECU shall not send a response message.

Table 171 — 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	02_{16}	SIDRQ
#2	PID: DTC that caused required freeze frame data storage	02_{16}	PID
#3	Frame #	00_{16}	FRNO

Table 172 — 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	42_{16}	SIDRQ
#2	PID: DTC that caused required freeze frame data storage	02_{16}	PID
#3	Frame #	00_{16}	FRNO
#4	DTC High Byte of P0000 {no freeze frame data stored}	00_{16}	DATA_A
#5	DTC Low Byte of P0000 {no freeze frame data stored}	00_{16}	DATA_B

Case #3: Multiple freeze frames

Cases #1 and #2 imply a scenario where only the required freeze frame (frame 00_{16}) is stored. This scenario implies the use of static PID support data where PID support data for a given ECU does not change for different frames or different DTCs. Since the PID support data are static, it can be obtained even before a freeze frame is stored.

Manufacturers who wish to store multiple freeze frames or, where allowed by OBD regulations, who wish to store different PID data in freeze frame based on the DTC would be required to use dynamic PID support data. Dynamic PID support data allows for different PID support data for different freeze frames and for different DTCs. Because of this, dynamic PID support data are not valid until a freeze frame for a particular frame has been stored. Requesting PID support data before a freeze frame is stored would indicate that only PID 02_{16} is supported.

External test equipment that supports dynamic PID support data for freeze frame retrieval will be compatible with ECUs that support static PID support data as well as dynamic PID support data and is therefore the recommended approach.

In this example, every freeze frame supports a different set of PIDs. PID support cannot be determined until after a freeze frame is stored. In order to determine if there are any frames stored, the external test equipment shall request PID 02_{16} of freeze frame 00_{16} from the vehicle, then request PID 02_{16} frame 01_{16}, then request PID 02_{16} frame 02_{16}, etc. Any frames that report a DTC will have freeze frame data stored. When a frame reports 0000_{16}, indicating no DTC stored and no freeze frame data, subsequent frames shall also report 0000_{16}. Note that this requires the ECU to store freeze frames in ascending order starting with frame 00_{16}, then 01_{16}, etc. There can be no gaps in the frame numbers, e.g. 00_{16}, then 02_{16}, then 05_{16}. If there are gaps, the tool would have to ask for every possible frame from 00_{16} to FF_{16} to make sure that all frames are available to the technician. Therefore, gaps are not allowed.

Next, the external test equipment presents a list of available DTCs to the technician. After the technician selects a DTC, the external test equipment requests the supported PIDs for the DTC the technician selected. Once the PIDs supported by that freeze frame have been determined, the external test equipment requests the supported PIDs for the frame associated with the DTC.

8.3 Service 03_{16} — Request emission-related diagnostic trouble codes

8.3.1 Functional description

The purpose of this service is to enable the external test equipment to obtain “confirmed” emission-related DTCs.

Send a Service 03_{16} request for all emission-related DTCs. Each ECU that has DTCs shall respond with one (1) message containing all emission-related DTCs. If an ECU does not have emission-related DTCs, then it shall respond with a message indicating no DTCs are stored by setting the parameter # of DTC to 00_{16}.

DTCs are transmitted in two (2) bytes of information for each DTC. The first two (2) bits (high order) of the first (1) byte for each DTC indicate whether the DTC is a powertrain, chassis, body, or network DTC (refer to SAE J2012 for additional interpretation of this structure). The second two (2) bits shall indicate the first digit of the DTC (0 through 3). The second (2) nibble of the first (1) byte and the entire second (2) byte are the next three (3) hexadecimal characters of the actual DTC reported as hexadecimal. A powertrain DTC transmitted as 0143_{16} shall be displayed as P0143.

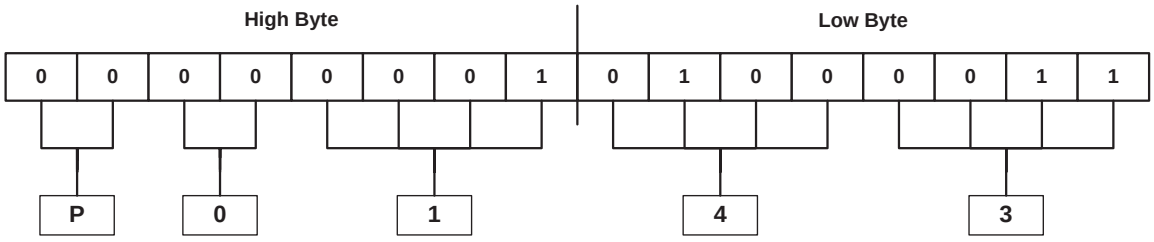


Figure 19 — Diagnostic trouble code encoding example DTC P0143

8.3.2 Message data bytes

8.3.2.1 Request emission-related DTC request message definition

Table 173 — Request emission-related DTC request message

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request emission-related DTC request SID	M	03_{16}	SIDRQ

8.3.2.2 Request emission-related DTC response message definition

Table 174 — Request emission-related DTC response message

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request emission-related DTC response SID	M	43_{16}	SIDPR
#2	# of DTC = [no emission-related DTCs stored emission-related DTCs stored]	M	XX_{16} = [00_{16}, 01_{16} – FF_{16}	#OFDTC
#3 #4	DTC#1 (High Byte) DTC#1 (Low Byte)	C[a] C	XX_{16} XX_{16}	DTC1HI DTC1LO
:	:	:	XX_{16}	
#n-1 #n	DTC#m (High Byte) DTC#m (Low Byte)	C C	XX_{16} XX_{16}	DTCmHI DTCmLO

[a] C = Conditional. DTC#1 - DTC#m are only included if # of DTC parameter value ≠ 00_{16}.

8.3.3 Parameter definition

The # of DTC parameter reports the emission-related DTC(s) currently (at the time of the request message processing) stored in the ECU(s).

8.3.4 Message example

The tables below show 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 175 — 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**	**Mnemonic**
#1	Request emission-related DTCs request SID	03_{16}	SIDRQ

Table 176 — 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**	**Mnemonic**
#1	Request emission-related DTCs response SID	43_{16}	SIDPR
#2	# of DTC {number of emission-related DTCs stored in this ECU}	06_{16}	#OFDTC
#3	DTC High Byte of P0143	01_{16}	DTC1HI
#4	DTC Low Byte of P0143	43_{16}	DTC1LO
#5	DTC High Byte of P0196	01_{16}	DTC2HI
#6	DTC Low Byte of P0196	96_{16}	DTC2LO
#7	DTC High Byte of P0234	02_{16}	DTC3HI
#8	DTC Low Byte of P0234	34_{16}	DTC3LO
#9	DTC High Byte of P02CD	02_{16}	DTC4HI
#10	DTC Low Byte of P02CD	CD_{16}	DTC4LO
#11	DTC High Byte of P0357	03_{16}	DTC5HI
#12	DTC Low Byte of P0357	57_{16}	DTC5LO
#13	DTC High Byte of P0A24	$0A_{16}$	DTC6HI
#14	DTC Low Byte of P0A24	24_{16}	DTC6LO

Table 177 — 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**	**Mnemonic**
#1	Request emission-related DTCs response SID	43_{16}	SIDPR
#2	# of DTC {number of emission-related DTCs stored in this ECU}	00_{16}	#OFDTC

Table 178 — 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**	**Mnemonic**
#1	Request emission-related DTCs response SID	43_{16}	SIDPR
#2	# of DTC {number of emission-related DTCs stored in this ECU}	01_{16}	#OFDTC
#3	DTC High Byte of P0443	04_{16}	DTC1HI
#4	DTC Low Byte of P0443	43_{16}	DTC1LO

8.4 Service 04_{16} — Clear/Reset emission-related diagnostic information

8.4.1 Functional description

The purpose of this service is to provide a means for the external test equipment to command ECUs to clear all emission-related diagnostic information. This includes the following:

- MIL and number of diagnostic trouble codes (can be read with Service 01_{16}, PID 01_{16});
- clear the I/M (Inspection/Maintenance) readiness bits (can be read with Service 01_{16}, PID 01_{16});
- confirmed diagnostic trouble codes (can be read with Service 03_{16});
- pending diagnostic trouble codes (can be read with Service 07_{16});
- diagnostic trouble code for freeze frame data (can be read with Service 02_{16}, PID 02_{16});
- freeze frame data (can be read with Service 02_{16});
- status of system monitoring tests (can be read with Service 01_{16}, PID 41_{16});
- on-board monitoring test results (can be read with Service 06_{16});
- distance travelled while MIL is activated (can be read with Service 01_{16}, PID 21_{16});
- number of warm-ups since DTCs cleared (can be read with Service 01_{16}, PID 30_{16});
- distance travelled since DTCs cleared (can be read with Service 01_{16}, PID 31_{16});
- engine run time while MIL is activated (can be read with Service 01_{16}, PID $4D_{16}$);
- engine run time since DTCs cleared (can be read with Service 01_{16}, PID $4E_{16}$);
- reset misfire counts of Standardized Test ID $0B_{16}$ to zero (can be read with Service 06_{16}).

Other manufacturer-specific “clearing/resetting” actions may also occur in response to this request message. All ECUs shall respond to this request message with ignition ON and with the engine not running.

For safety and/or technical design reasons, ECUs that cannot perform this operation under other conditions, such as with the engine running, shall send a negative response message with NRC 22_{16} - conditionsNotCorrect.

Some OBD regulations may require that all OBD ECUs clear diagnostic information under the same conditions (all ECUs shall clear diagnostic information with the engine off). If one ECU cannot clear diagnostic information with the engine running, then all OBD ECUs are required to respond in the same manner and not clear diagnostic information with the engine running.

8.4.2 Message data bytes

8.4.2.1 Clear/reset emission-related diagnostic information request message definition

Table 179 — Clear/reset emission-related diagnostic information request message

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Clear/reset emission-related diagnostic information request SID	M	04_{16}	SIDRQ

8.4.2.2 Clear/reset emission-related diagnostic information response message definition

Table 180 — Clear/reset emission-related diagnostic information response message

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Clear/reset emission-related diagnostic information response SID	M	44_{16}	SIDPR

8.4.3 Parameter definition

This service does not support any parameters.

8.4.4 Message example

The example below shows how the “clear/reset emission-related diagnostic information” service shall be implemented if ignition is ON and the engine is not running. The external test equipment commands the vehicle to “clear/reset emission-related diagnostic information”.

Table 181 — 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**	**Mnemonic**
#1	Clear/reset emission-related diagnostic information request SID	04_{16}	SIDRQ

Table 182 — 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**	**Mnemonic**
#1	Clear/reset emission-related diagnostic information response SID	44_{16}	SIDPR

Table 183 — 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**	**Mnemonic**
#1	Clear/reset emission-related diagnostic information response SID	44_{16}	SIDPR

Table 160 shows a negative response to “clear/reset emission-related diagnostic information” for an ECU that cannot clear diagnostic information with the engine running.

Table 184 — Negative response message

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Negative Response Service Identifier	$7F_{16}$	SIDNR
#2	Clear/reset emission-related diagnostic information request SID	04_{16}	SIDRQ
#3	Negative Response Code: conditionsNotCorrect	22_{16}	NR_CNC

8.5 Service 05_{16} — Request oxygen sensor monitoring test results

Service 05_{16} is not supported for ISO 15765-4. The functionality of Service 05_{16} is implemented in Service 06_{16}.

8.6 Service 06_{16} — Request on-board monitoring test results for specific monitored systems

8.6.1 Functional description

The purpose of this service is to allow access to the results for On-Board Diagnostic monitoring tests of specific components/systems that are continuously monitored (e.g. misfire monitoring for gasoline vehicles) and non-continuously monitored (e.g. catalyst system).

The request message for test values includes an On-Board Diagnostic Monitor ID (see SAE J1979-DA) that indicates the information requested. The response message for test values includes Unit and Scaling information which is defined in SAE J1979-DA. The vehicle manufacturer shall use Unit and Scaling IDs that most closely match the physical quantities used for monitoring in order to make the information more useful to a service technician for diagnostic purposes, e.g. an On-Board Diagnostic Monitor ID in which the monitor checks for a pressure change shall utilize a Unit and Scaling ID which includes pressure in the description.

The vehicle manufacturer is responsible for assigning “Manufacturer Defined Test IDs” for different tests of a monitored system. The latest valid test values (results) are to be retained, even over multiple ignition OFF cycles, until replaced by more recent test values (results). Test values (results) are requested by On-Board Diagnostic Monitor ID. Test values (results) are always reported with the Minimum and Maximum Test Limits. The Unit and Scaling ID included in the response message defines the scaling and unit to be used by the external test equipment to display the test values (results), Minimum Test Limit, and Maximum Test Limit information.

If an On-Board Diagnostic Monitor has not been completed at least once since a “clear/reset emission-related diagnostic information” request was carried out or battery disconnect that erased the latest valid test values, then the parameters Test Value (Results), Minimum Test Limit, and Maximum Test Limit shall be set to zero (0000_{16}) values. Note that for some unit and scaling IDs, 0000_{16} translates to a non-zero result (e.g. Unit and Scaling ID 16_{16} for temperature, 0000_{16} displays as −40.0 °C) so some monitors that have not completed may show test results, minimum limits, and maximum limits that, after scaling, are all equal but are non-zero.

Not all On-Board Diagnostic Monitor IDs are applicable or supported by all systems. On-Board Diagnostic Monitor ID 00_{16} is a bit-encoded value that indicates for each ECU which On-Board Diagnostic Monitor IDs are supported. On-Board Diagnostic Monitor ID 00_{16} indicates support for On-Board Diagnostic Monitor IDs from 01_{16} to 20_{16}. On-Board Diagnostic Monitor ID 20_{16} indicates support for On-Board Diagnostic Monitor IDs 21_{16} through 40_{16}, etc. This is the same concept for PIDs/TIDs/INFOTYPEs support in Services 01_{16}, 02_{16}, 06_{16}, 08_{16}, and 09_{16}. On-Board Diagnostic Monitor ID 00_{16} is required for those ECUs that respond to a corresponding Service 06_{16} request message as specified in SAE J1979-DA.

The request message including supported On-Board Diagnostic Monitor IDs may contain up to six (6) OBDMIDs. A request message including an On-Board Diagnostic Monitor ID, which reports test values shall only contain one (1) OBDMID. External test equipment shall not request a combination of OBDMIDs supported and a single OBDMID, which report test values. The ECU shall support requests for up to six (6) supported OBDMIDs and only one (1) OBDMID which reports test values.

A unique method shall be utilized for displaying data for monitors that have multiple tests. Many OBD monitors have multiple tests that are done in either a serial or parallel manner. If a monitor uses multiple OBD Monitor ID/Test ID combinations that may not all complete at the same time, the following method shall be used to update the stored test results at the time of monitor completion.

After the monitor completes, update all Monitor ID/Test ID combinations (or "test results") that were utilized by the monitor with appropriate passing or failing results. If a test result (or "Monitor ID/Test ID") was not utilized during this monitoring event, set the Test Values and Minimum and Maximum Test Limits to their initial values (0000_{16}, test not completed). Test results from the previously completed monitoring events shall not be mixed with test results from the current completed monitoring event.

In some cases, test results (or "Monitor ID/Test ID combinations") will be displayed as being incomplete even though the monitor (as indicated by PID 41_{16}) was successfully completed and either passed or failed. In other cases, some Test IDs will show passing results while others will show failing results after the monitor (as indicated by PID 41_{16}) was successfully completed and failed. Note that OBD-II regulations prohibit a passing monitor from showing any failing test results. If an initial serial test indicates a failure and a subsequent re-test of the system indicates a passing result, the test that was utilized to make the passing determination should be displayed while the failing test that was utilized to make the initial determination should be reset to its initial values (0000_{16}, test not completed).

An example for a serial monitor is an evaporative leakage monitor where the monitor first checks a reference leak and then starts to execute the actual leakage check. If the reference test fails, then the leakage test is not executed.

As an example of a parallel monitor, a purge valve flow monitor can pass by having a large rich lambda shift, a large lean lambda shift, or a large engine rpm increase. If the purge valve is activated and a large rich lambda shift occurs, the Test ID for the rich lambda shift would show a passing result while the other two Test IDs would show incomplete. Since some Test IDs for a completed monitor will show incomplete, PID 41_{16} shall be used to determine monitor completion status.

8.6.2 Message data bytes

8.6.2.1 Request on-board monitoring test results for specific monitored systems request message definition (read-supported OBDMIDs)

Table 185 — Request on-board monitoring test results for specific monitored systems request message (read-supported OBDMIDs)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request on-board monitoring test results for specific monitored systems request SID	M	06_{16}	SIDRQ
#2	On-Board Diagnostic Monitor ID (OBDMIDs supported: SAE J1979-DA)	M	XX_{16}	OBDMID
#3	On-Board Diagnostic Monitor ID (OBDMIDs supported: SAE J1979-DA)	U[a]	XX_{16}	OBDMID
#4	On-Board Diagnostic Monitor ID (OBDMIDs supported: SAE J1979-DA)	U	XX_{16}	OBDMID
#5	On-Board Diagnostic Monitor ID (OBDMIDs supported: SAE J1979-DA)	U	XX_{16}	OBDMID

[a] U = User Optional. OBDMID may be included to avoid multiple OBDMID supported request messages.

Table 185 *(continued)*

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#6	On-Board Diagnostic Monitor ID (OBDMIDs supported: SAE J1979-DA)	U	XX_{16}	OBDMID
#7	On-Board Diagnostic Monitor ID (OBDMIDs supported: SAE J1979-DA)	U	XX_{16}	OBDMID

[a] U = User Optional. OBDMID may be included to avoid multiple OBDMID supported request messages.

To request OBDMIDs supported range from $C1_{16}$ - FF_{16}, another request message with OBDMID#1 = $C0_{16}$ and OBDMID#2 = $E0_{16}$ shall be sent to the vehicle.

8.6.2.2 Request on-board monitoring test results for specific monitored systems response message definition (report supported OBDMIDs)

ECU(s) shall respond to all supported ranges if requested. A range is defined as a block of 32 OBDMIDs (e.g. range #1: OBDMIDs 01_{16} - 20_{16}). The ECU shall not respond to unsupported OBDMID ranges unless subsequent ranges have a supported OBDMID(s).

Table 186 — Request on-board monitoring test results for specific monitored systems response message (report supported OBDMIDs)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request on-board monitoring test results for specific monitored systems response SID	M	46_{16}	SIDPR
	data record of supported OBDMID = [			OBDMIDREC
#2	1st supported OBDMID	M	XX_{16}	OBDMID
#3	Data A: supported OBDMIDs,	M	XX_{16}	DATA_A
#4	Data B: supported OBDMIDs,	M	XX_{16}	DATA_B
#5	Data C: supported OBDMIDs,	M	XX_{16}	DATA_C
#6	Data D: supported OBDMIDs]	M	XX_{16}	DATA_D
:	:	:	:	:
	data record of supported OBDMID = [			OBDMIDREC
#n-4	mth supported OBDMID	C1[a]	XX_{16}	OBDMID
#n-3	Data A: supported OBDMIDs,	C2[b]	XX_{16}	DATA_A
#n-2	Data B: supported OBDMIDs,	C2	XX_{16}	DATA_B
#n-1	Data C: supported OBDMIDs,	C2	XX_{16}	DATA_C
#n	Data D: supported OBDMIDs]	C2	XX_{16}	DATA_D

[a] C1 = Conditional. OBDMID value shall be the same value as included in the request message if supported by the ECU.
[b] C2 = Conditional. Value indicates OBDMIDs supported; range of supported OBDMIDs depends on selected OBDMID value (see C1).

The response message shall only include the OBDMID(s) and Data A to D, which are supported by the ECU. If the request message includes an OBDMID value(s) which is (are) not supported by the ECU, those shall not be included in the response message.

8.6.2.3 Request on-board monitoring test results for specific monitored systems request message definition (read OBDMID test values)

Table 187 — Request on-board monitoring test results for specific monitored systems request message (read OBDMID test values)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request on-board monitoring test results for specific monitored systems request SID	M	06_{16}	SIDRQ
#2	On-Board Diagnostic Monitor ID	M	XX_{16}	OBDMID

8.6.2.4 Request on-board monitoring test results for specific monitored systems response message definition (report OBDMID test values)

Table 188 — Request on-board monitoring test results for specific monitored systems response message (report OBDMID test values)

Data Byte	Parameter Name		Cvt	Byte Value	Mnemonic
#1	Request on-board monitoring test results for specific monitored systems response SID		M	46_{16}	SIDPR
	data record of supported OBDMID = [				OBDMIDREC
#2		On-Board Diagnostic Monitor ID	M	XX_{16}	OBDMID
#3		Std./Manuf. Defined TID#1	M	XX_{16}	S/MDTID
#4		Unit And Scaling ID#1	M	XX_{16}	UASID
#5		Test Value (High Byte)#1	M	XX_{16}	TVHI
#6		Test Value (Low Byte)#1	M	XX_{16}	TVLO
#7		Min Test Limit (High Byte)#1	M	XX_{16}	MINTLHI
#8		Min Test Limit (Low Byte)#1	M	XX_{16}	MINTLLO
#9		Max Test Limit (High Byte)#1	M	XX_{16}	MAXTLHI
#10		Max Test Limit (Low Byte)#1]	M	XX_{16}	MAXTLLO
:	:		:	:	:
	data record of supported OBDMID = [				OBDMIDREC
#n-8		On-Board Diagnostic Monitor ID	C1[a]	XX_{16}	OBDMID
#n-7		Std./Manuf. Defined TID#m	C2[b]	XX_{16}	S/MDTID
#n-6		Unit And Scaling ID#m	C2	XX_{16}	UASID
#n-5		Test Value (High Byte)#m	C2	XX_{16}	TVHI
#n-4		Test Value (Low Byte)#m	C2	XX_{16}	TVLO
#n-3		Min Test Limit (High Byte)#m	C2	XX_{16}	MINTLHI
#n-2		Min Test Limit (Low Byte)#m	C2	XX_{16}	MINTLLO
#n-1		Max Test Limit (High Byte)#m	C2	XX_{16}	MAXTLHI
#n		Max Test Limit (Low Byte)#m]	C2	XX_{16}	MAXTLLO

[a] C1 = Conditional. Parameter is only present if more than one (1) Manufacturer Defined TID is supported by the ECU for the requested Monitor ID.

[b] C2 = Conditional. Parameter and value depend on selected Manufacturer Defined TID number and are only included if the Manufacturer Defined TID is supported by the ECU. The value shall be zero (00) in case the On-Board Diagnostic Monitor has not been completed at least once since clear/reset emission-related diagnostic information or battery disconnect.

8.6.3 Parameter definition

8.6.3.1 On-Board Diagnostic Monitor IDs supported

The On-Board Diagnostic Monitor IDs supported is the same concept as used for PID support in Services 01_{16} and 02_{16} as specified in SAE J1979-DA.

8.6.3.2 On-Board Diagnostic Monitor ID description

The On-Board Diagnostic Monitor ID is a one (1) byte parameter and is defined in SAE J1979-DA. An On-Board Diagnostic Monitor may have more than one (1) monitor test (Test ID).

NOTE The On-Board Diagnostic Monitor ID is similar to the Test ID parameter specified in Service 06_{16} in 7.6.3.2.

8.6.3.3 Standardized and Manufacturer Defined Test ID description

The Standardized and Manufacturer Defined Test ID is a one (1) byte parameter. For example, the On-Board Diagnostic Monitor “Oxygen Sensor Monitor Bank 1 - Sensor 1” or the On-Board Diagnostic Misfire Monitor may use some of the following Standardized Test IDs.

Test IDs used in Service 05_{16} which are constants (01_{16}, 02_{16}, 03_{16}, 04_{16}) are not required to be supported in Service 06_{16}.

Table 165 describes Standardized Test IDs.

For the Standardized Test IDs that are constant values, the Minimum and Maximum Test Limits shall be the same values as reported for the Test Value.

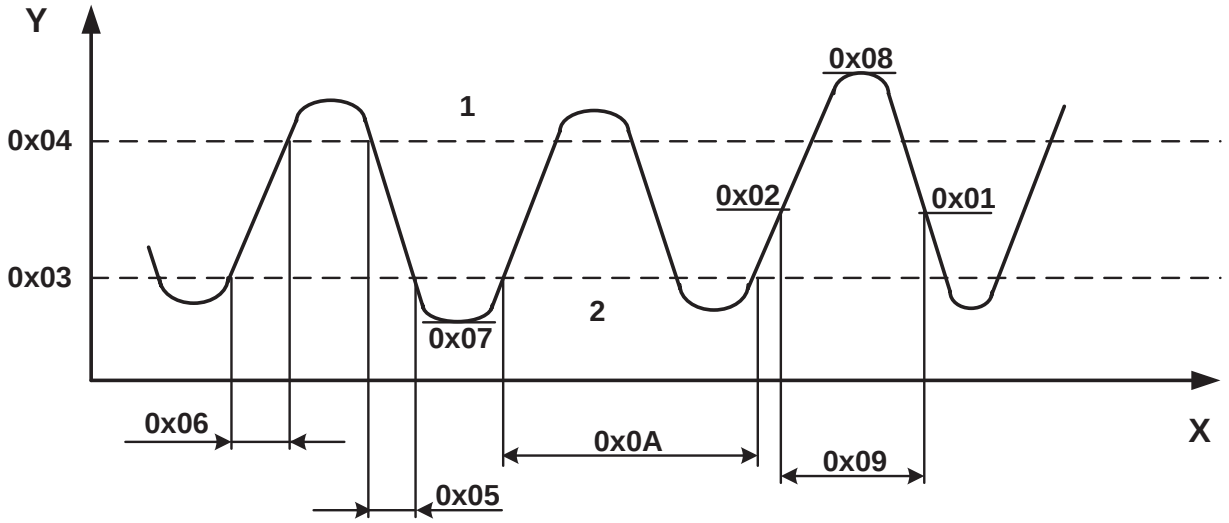
Table 189 — Standardized Test ID description

Range	Description
00_{16}	ISO/SAE reserved
01_{16}	Rich to lean sensor threshold voltage (constant)
02_{16}	Lean to rich sensor threshold voltage (constant)
03_{16}	Low sensor voltage for switch time calculation (constant)
04_{16}	High sensor voltage for switch time calculation (constant)
05_{16}	Rich to lean sensor switch time (calculated)
06_{16}	Lean to rich sensor switch time (calculated)
07_{16}	Minimum sensor voltage for test cycle (calculated)
08_{16}	Maximum sensor voltage for test cycle (calculated)
09_{16}	Time between sensor transitions (calculated)
$0A_{16}$	Sensor period (calculated)
$0B_{16}$	Exponential Weighted Moving Average (EWMA) misfire counts for previous driving cycles (calculated, rounded to an integer value) General EWMA calculation: 0,1 * (current misfire counts) + 0,9 * (previous misfire counts average) Initial value for (previous misfire counts average) = 0 Internal ECU calculation registers with precision higher than one count shall be used and retained to calculate the contents of registers $0B_{16}$ and $0C_{16}$ to prevent rounding errors. If this is not done, these registers will never count back down to zero after misfire stops. The calculations shall be carried out using the high-precision registers, then rounded to the nearest integer value to be output as register $0B_{16}$ and $0C_{16}$. High_Precision_EWMA_Misfire_Counts $_{current}$ = Rounded [(0,1) * High_Precision_Misfire_Counts $_{current}$ + (0,9) * High_Precision_EWMA_Misfire_Counts $_{previous}$] Where: Rounded means rounded to the nearest integer. The high-precision values are never reported, they are only used for internal calculations. This Test ID shall be reported with OBD Monitor IDs $A2_{16}$ – AD_{16} (refer to SAE J1979-DA) and the Scaling ID 24_{16} (refer to SAE J1979-DA).
$0C_{16}$	Misfire counts for last/current driving cycles (calculated, rounded to an integer value). This Test ID shall be reported with OBD Monitor IDs $A2_{16}$ – AD_{16} (see SAE J1979-DA) and the Scaling ID 24_{16} (see SAE J1979-DA).
$0D_{16}$ –$7F_{16}$	Reserved for future standardization

Table 190 — Manufacturer Defined Test ID description

Range	Description
80_{16} – FE_{16}	Manufacturer Defined Test ID range — This parameter is an identifier for the test performed within the On-Board Diagnostic Monitor.
FF_{16}	ISO/SAE reserved

The results of the latest mandated on-board oxygen sensor monitoring tests can be seen in Figure 20.



Key

1 rich

2 lean

Figure 20 — Standardized Test ID value example

8.6.3.4 Example for use of Standardized Test IDs for misfire monitor

OBD regulations may require reporting the number of misfires detected during the current driving cycle (Test ID $0C_{16}$) and the average number of misfires detected during the previous driving cycles (Test ID $0B_{16}$) for each cylinder. Therefore, for a four-cylinder engine, eight (8) pieces of data shall be reported for both Test IDs. The purpose of the misfire data is to help a service technician identify which cylinders are currently misfiring ($0C_{16}$) and identify which cylinders have been consistently misfiring in the previous driving cycles ($0B_{16}$). The actual misfire counts will depend on how the vehicle was driven, how long it was driven, etc. Misfire counts for cylinders shall only be compared relative to each other. If some cylinders have many more misfires than other cylinders, the technician should probably begin his troubleshooting with the cylinders that have the highest misfire counts.

The $0B_{16}$ registers contain the Exponential Weighted Moving Average (EWMA) values for misfire counted during previous driving cycles. The EWMA values should only be recalculated once per driving cycle. This calculation can be done every power-up or every power-down sequence if the ECU stays alive after the ignition key is turned off. The EWMA value uses the misfire counts collected during the last/current driving cycle. The value of the $0C_{16}$ counters, after the driving cycle ends, is the number of misfires counted during the current/last driving cycle. The software shall take the contents of the $0B_{16}$ register (this is the previous average) multiply by 0,9 and add the contents of the $0C_{16}$ register (this is the current counts) multiplied by 0,1. This becomes the new EWMA value.

The internal ECU calculation registers with precision higher than one count shall be used and retained to calculate the contents of registers $0B_{16}$ and $0C_{16}$ to prevent rounding errors. If this is not done, these registers will never count back down to zero after misfire stops. The calculations shall be done using the high-precision registers, then rounded to the nearest integer value to be output as register $0B_{16}$ and $0C_{16}$. The last row of Table 167 shows the high-precision internal calculation.

The Test ID $0C_{16}$ counters shall count misfires for each cylinder and save them in keep alive or non-volatile memory. They should update continuously, in 200 or 1 000 revolution increments, as a minimum. When the engine starts, the $0C_{16}$ misfire counters shall be reset to zero. Prior to engine start-up, the last value from the previous driving cycle shall be retained and displayed until the engine starts so that a service technician can see how many misfires occurred the last time the vehicle was driven.

If a vehicle has constant misfire in one or more cylinders, the service technician can watch the Test ID $0C_{16}$ counters count-up as he drives the vehicle, up to a maximum of 65 535 misfires. If the technician

is driving and watching the $0C_{16}$ counters, he would be seeing misfire counts for the "current" driving cycle. If he turns off the ignition key, he has just ended the current driving cycle. If he then turns the key back on but does not start the engine, the $0C_{16}$ counters will contain the number of misfires that occurred during the "last" driving cycle. If the technician now starts the engine, the $0C_{16}$ counters will be reset to zero and the software starts counting misfires all over again.

There are no minimum or maximum misfire monitor threshold limits for misfire counts. Test IDs $0B_{16}$ and $0C_{16}$ just accumulate the number of misfires that occurred. These counts should accumulate with or without a misfire DTC. If there was a little misfire but not enough to store a DTC, Test ID $0B_{16}$ and $0C_{16}$ values for each cylinder should still show the number of misfires that occurred. The minimum test limit value should be 0; the maximum test limit value should be 65 535. Therefore, there will never be a "fail" result.

For this example, the vehicle PCM or ECM does not stay alive after shutdown so EWMA values are updated at every power-up.

Table 191 — Misfire Test ID $0B_{16}$ and $0C_{16}$ example

Misfire counts	**Cyl #1 Counts**	**Cyl #1 EWMA**	**Cyl #2 Counts**	**Cyl #2 EWMA**	**Cyl#3 Counts**	**Cyl#3 EWMA**	**Cyl#4 Counts**	**Cyl#4 EWMA**
Monitor ID/Test ID	$A2_{16}$/ $0C_{16}$	$A2_{16}$/ $0B_{16}$	$A3_{16}$/ $0C_{16}$	$A3_{16}$/ $0B_{16}$	$A4_{16}$/ $0C_{16}$	$A4_{16}$/ $0B_{16}$	$A5_{16}$/ $0C_{16}$	$A5_{16}$/ $0C_{16}$
Key on, drive cycle 1	0	0	0	0	0	0	0	0
Start engine	0	0	0	0	0	0	0	0
Drive with misfire	200	0	1	0	500	0	9	0
Key off	200	0	1	0	500	0	9	0
Key on, drive cycle 2	200	20	1	0	500	50	9	1
Start engine	0	20	0	0	0	50	0	1
Drive with misfire	1 000	20	4	0	3 000	50	12	1
Key off	1 000	20	4	0	3 000	50	12	1
Key on, drive cycle 3	1 000	118	4	0	3 000	345	12	2
Start engine	0	118	0	0	0	345	0	2
Drive with misfire	1 000	118	4	0	3 000	345	12	2
Key off	1 000	118	4	0	3 000	345	12	2
Key on, drive cycle 4	1 000	206	4	0	3 000	611	12	3
Start engine	0	206	0	0	0	611	0	3
Drive with misfire	1 000	206	4	0	3 000	611	12	3
Key off	1 000	206	4	0	3 000	611	12	3
Key on, drive cycle 5	1 000	286	4	0	3 000	849	12	4
Start engine	0	286	0	0	0	849	0	4
Drive with misfire	1 000	286	4	0	3 000	849	12	4
Key off	1 000	285	4	0	3 000	849	12	4
Key on, drive cycle 6	1 000	357	4	0	3 000	1 065	12	5
Start engine	0	357	0	0	0	1 065	0	5
Drive with misfire	1 000	357	4	0	3 000	1 065	12	5
Key off	1 000	357	4	0	3 000	1 065	12	5
Key on, drive cycle 12	1 000	692	4	0	3 000	2 074	12	8
Start engine	0	692	0	0	0	2 074	0	8

Table 191 *(continued)*

Misfire counts	Cyl #1 Counts	Cyl #1 EWMA	Cyl #2 Counts	Cyl #2 EWMA	Cyl#3 Counts	Cyl#3 EWMA	Cyl#4 Counts	Cyl#4 EWMA
Monitor ID/ Test ID	$A2_{16}$/ $0C_{16}$	$A2_{16}$/ $0B_{16}$	$A3_{16}$/ $0C_{16}$	$A3_{16}$/ $0B_{16}$	$A4_{16}$/ $0C_{16}$	$A4_{16}$/ $0B_{16}$	$A5_{16}$/ $0C_{16}$	$A5_{16}$/ $0C_{16}$
Drive with misfire	1 000	692	4	0	3 000	2 074	12	8
Key off	1 000	692 (692,456)	4	0 (0,444)	3 000	2 074 (2 074,259)	12	8 (8,130)

8.6.3.5 Unit and Scaling ID definition

The Unit and Scaling ID is a one (1) byte identifier to reference the scaling and unit to be used by the external test equipment to calculate and display the test values (results), Minimum Test Limit, and the Maximum Test Limit for the Standardized and Manufacturer Defined Test ID requested. All standardized Unit and Scaling IDs are specified in SAE J1979-DA.

8.6.3.6 Test Value (Result) description

The Test Value represents the test result and is defined in Table 168.

Table 192 — Test Value description

Parameter Name	# of Bytes	Description
Test Value	2 (High and Low Byte)	Test Value (Result) — This value shall be calculated and displayed by the external test equipment based on the Unit and Scaling ID included in the response message. The Test Value shall be within the Minimum and Maximum Test Limit to indicate a "Pass" result.

8.6.3.7 Minimum Test Limit description

The Minimum Test Limit parameter is defined in Table 169.

Table 193 — Minimum Test Limit description

Parameter Name	# of Bytes	Description
Minimum Test Limit	2 (High and Low Byte)	The Minimum Test Limit shall be calculated and displayed by the external test equipment based on the Unit and Scaling ID included in the response message. The Unit and Scaling IDs are specified in SAE J1979-DA of this document monitor identified by the On-Board Diagnostic Monitor ID. For the Standardized Test IDs that are constant values, the Minimum Test Limit shall be the same value as reported for the Test Value. The following conditions apply: — if the Test Value is less than the Minimum Test Value, this results in a "Fail" condition; — if the Test Value equals the Minimum Test Value, this results in a "Pass" condition; — if the Test Value is greater than the Minimum Test Value, this results in a "Pass" condition.

8.6.3.8 Maximum Test Limit description

The Maximum Test Limit parameter is defined in Table 170.

Table 194 — Maximum Test Limit description

Parameter Name	# of Bytes	Description
Maximum Test Limit	2 (High and Low Byte)	The Maximum Test Limit shall be calculated and displayed by the external test equipment based on the Unit and Scaling ID included in the response message. The Unit and Scaling IDs are specified in SAE J1979-DA. The Maximum Test Limit shall be the maximum value for the monitor identified by the On-Board Diagnostic Monitor ID. For the Standardized Test IDs that are constant values, the Maximum Test Limit shall be the same value as reported for the Test Value. The following conditions apply: — if the Test Value is less than the Maximum Test Value, this results in a "Pass" condition; — if the Test Value equals the Maximum Test Value, this results in a "Pass" condition; — if the Test Value is greater than the Maximum Test Value, this results in a "Fail" condition.

8.6.4 Message example

Tables 171 and 172 show how the "request on-board monitoring test results for specific monitored systems" service shall be implemented.

8.6.4.1 Step #1: Request on-board monitoring test results for specific monitored systems (request for supported OBDMIDs)

The external test equipment requests all supported OBDMIDs from the vehicle. Refer to the example of Service 01_{16} for guidance in requesting supported PIDs (the same concept is used for supported OBDMIDs).

As a result of the supported OBDMID request, the external test equipment creates an internal list of supported OBDMIDs for each ECU. ECU#1 (ECM) supports OBDMIDs 01_{16}, 05_{16}, 10_{16}, and 21_{16}. ECU#2 (TCM) does not support any OBDMIDs.

8.6.4.2 Step #2: Request on-board monitoring test results for specific monitored systems

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 01_{16} - Oxygen Sensor Monitor Bank 1 - Sensor 1

Table 195 — 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**	**Mnemonic**
#1	Request on-board monitoring test results for specific monitored systems request SID	06_{16}	SIDRQ
#2	OBDMID: 01_{16} - Oxygen Sensor Monitor Bank 1 - Sensor 1	01_{16}	OBDMID

Table 196 — 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**	**Mnemonic**
#1	Request on-board monitoring test results for specific monitored systems response SID	46_{16}	SIDPRQ
#2	OBDMID: 01_{16} - Oxygen Sensor Monitor Bank 1 - Sensor 1	01_{16}	OBDMID
#3	Standardized Test ID: 01_{16} - Rich to lean sensor threshold voltage (constant)	01_{16}	STID
#4	Unit And Scaling ID: Voltage	$0A_{16}$	UASID
#5	Test Value High Byte:	$0B_{16}$	TESTVAL
#6	Test Value Low Byte: 0,365 V	$B0_{16}$	TESTVAL
#7	Minimum Test Limit High Byte:	$0B_{16}$	MINLIMIT
#8	Minimum Test Limit Low Byte: 0,365 V	$B0_{16}$	MINLIMIT
#9	Maximum Test Limit High Byte:	$0B_{16}$	MAXLIMIT
#10	Maximum Test Limit Low Byte: 0,365 V	$B0_{16}$	MAXLIMIT
#11	OBDMID: 01 - Oxygen Sensor Monitor Bank 1 - Sensor 1	01_{16}	OBDMID
#12	Standardized Test ID: 05_{16} - Rich to lean sensor switch time (calculated)	05_{16}	STID
#13	Unit And Scaling ID: Time	10_{16}	UASID
#14	Test Value High Byte	00_{16}	TESTVAL
#15	Test Value Low Byte: 0,072 s (0 min, 0 s)	48_{16}	TESTVAL
#16	Minimum Test Limit High Byte	00_{16}	MINLIMIT
#17	Minimum Test Limit Low Byte: 0,000 s (0 min, 0 s)	00_{16}	MINLIMIT
#18	Maximum Test Limit High Byte	00_{16}	MAXLIMIT
#19	Maximum Test Limit Low Byte: 0,100 s (0 min, 0 s)	64_{16}	MAXLIMIT
#20	OBDMID: 01_{16} - Oxygen Sensor Monitor Bank 1 - Sensor 1	01_{16}	OBDMID
#21	Manufacturer Defined Test ID: 133_{dec} (The name of this Test ID shall be documented in the vehicle Service Information.)	85_{16}	MDTID
#22	Unit And Scaling ID: Counts	24_{16}	UASID
#23	Test Value High Byte	00_{16}	TESTVAL
#24	Test Value Low Byte: 150 counts	96_{16}	TESTVAL
#25	Minimum Test Limit High Byte	00_{16}	MINLIMIT
#26	Minimum Test Limit Low Byte: 75 counts	$4B_{16}$	MINLIMIT
#27	Maximum Test Limit High Byte	FF_{16}	MAXLIMIT
#28	Maximum Test Limit Low Byte: 65 535 counts	FF_{16}	MAXLIMIT

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

8.6.4.3 Request on-board monitoring test results for specific monitored systems

In this example, the requested monitor has not been completed since a clear/reset emission-related diagnostic information event. The request message includes the following OBDMID request message: OBDMID 21_{16} - Catalyst Monitor Bank 1.

Table 197 — Request Catalyst Monitor Bank 1 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**	**Mnemonic**
#1	Request on-board monitoring test results for specific monitored systems request SID	06_{16}	SIDRQ
#2	OBDMID: 21_{16} - Catalyst Monitor Bank 1	21_{16}	OBDMID

Table 198 — Request Catalyst Monitor Bank 1 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**	**Mnemonic**
#1	Request on-board monitoring test results for specific monitored systems response SID	46_{16}	SIDPRQ
#2	OBDMID: 21_{16} - Catalyst Monitor Bank 1	21_{16}	OBDMID
#3	Manufacturer Defined Test ID: 135_{dec}	87_{16}	MDTID
#4	Unit And Scaling ID: Percent	$2E_{16}$	UASID
#5	Test Value High Byte: Monitor not completed at least once since erasure	00_{16}	TESTVAL
#6	Test Value Low Byte: 0,00 %	00_{16}	TESTVAL
#7	Minimum Test Limit High Byte	00_{16}	MINLIMIT
#8	Minimum Test Limit Low Byte: 0,00 %	00_{16}	MINLIMIT
#9	Maximum Test Limit High Byte	00_{16}	MAXLIMIT
#10	Maximum Test Limit Low Byte: 0,00 %	00_{16}	MAXLIMIT

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

8.7 Service 07_{16} — Request emission-related diagnostic trouble codes detected during current or last completed driving cycle

8.7.1 Functional description

The purpose of this service is to enable the external test equipment to obtain “pending” diagnostic trouble codes detected during current or last completed driving cycle for emission-related components/systems. Service 07_{16} is required for all DTCs and is independent of Service 03_{16}. The intended use of this data is to assist the service technician after a vehicle repair and after clearing diagnostic information by reporting test results after a single driving cycle. If the test failed during the driving cycle, the DTC associated with that test shall be reported. Test results reported by this service do not necessarily indicate a faulty component/system. If test results indicate a failure after additional driving, then the MIL will be illuminated and a DTC will be set and reported with Service 03_{16}, indicating a faulty component/system. This service can always be used to request the results of the latest test, independent of the setting of a DTC.

Test results for these components/systems shall be reported in the same format as the DTCs in Service 03_{16} (see the functional description for Service 03_{16}).

8.7.2 Message data bytes

8.7.2.1 Request emission-related diagnostic trouble codes detected during current or last completed driving cycle request message definition

Table 199 — Request emission-related diagnostic trouble codes detected during current or last completed driving cycle request message

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request emission-related diagnostic trouble codes detected during current or last completed driving cycle request SID	M	07_{16}	SIDRQ

8.7.2.2 Request emission-related diagnostic trouble codes detected during current or last completed driving cycle response message definition

Table 200 — Request emission-related diagnostic trouble codes detected during current or last completed driving cycle response message

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request emission-related diagnostic trouble codes detected during current or last completed driving cycle response SID	M	47_{16}	SIDPR
#2	# of DTC = [no emission-related DTCs # of emission-related DTCs]	M	 00_{16} 01_{16} – FF_{16}	#OFDTC
#3 #4	DTC#1 (High Byte) DTC#1 (Low Byte)	C[a] C	XX_{16} XX_{16}	DTC1HI DTC1LO
:	:	:	XX_{16}	
#n-1 #n	DTC#m (High Byte) DTC#m (Low Byte)	C C	XX_{16} XX_{16}	DTCmHI DTCmLO

[a] C = Conditional. DTC#1 - DTC#m are only included if # of DTC parameter value ≠ 00_{16}.

8.7.3 Parameter definition

The # of DTC parameter reports the emission-related DTC(s) currently (at the time of the request message processing) stored in the ECU(s).

8.7.4 Message example

Refer to message example of Service 03_{16}.

8.8 Service 08_{16} — Request control of on-board system, test, or component

8.8.1 Functional description

The purpose of this service is to enable the external test equipment to control the operation of an on-board system, test, or component.

The data bytes will be specified, if necessary, for each Test ID in SAE J1979-DA, and will be unique for each Test ID.

Possible uses for these data bytes in the request message are to

— turn on-board system/test/component on,

— turn on-board system/test/component off, and

— cycle on-board system/test/component for “n” seconds.

Possible uses for these data bytes in the response message are to

— report system status, and

— report test results.

Not all TIDs are applicable or supported by all systems. TID 00_{16} is a bit-encoded value that indicates for each ECU which TIDs are supported. TID 00_{16} indicates support for TIDs from 01_{16} to 20_{16}. TID 20_{16} indicates support for TIDs 21_{16} through 40_{16}, etc. This is the same concept for PIDs/TIDs/INFOTYPEs support in Services 01_{16}, 02_{16}, 06_{16}, 08_{16}, 09_{16}. TID 00_{16} is required for those ECUs that respond to a corresponding Service 08_{16} request message as specified in SAE J1979-DA.

The order of the TIDs in the response message is not required to match the order in the request message.

The request message, including supported Test IDs, may contain up to six (6) Test IDs. A request message, including a Test ID with optional data, shall only contain one (1) Test ID. External test equipment is not allowed to request a combination of Test IDs supported and a single Test ID with optional data. The ECU shall support requests for up to six (6) supported Test IDs and only one (1) Test ID with optional data.

8.8.2 Message data bytes

8.8.2.1 Request control of on-board device request message definition (read-supported TIDs)

Table 201 — Request control of on-board device request message (read-supported TIDs)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request control of on-board device request SID	M	08_{16}	SIDRQ
#2	TID#1 (Test IDs supported: SAE J1979-DA)	M	XX_{16}	TID
#3	TID#2 (Test IDs supported: SAE J1979-DA)	U	XX_{16}	TID
#4	TID#3 (Test IDs supported: SAE J1979-DA)	U	XX_{16}	TID
#5	TID#4 (Test IDs supported: SAE J1979-DA)	U	XX_{16}	TID
#6	TID#5 (Test IDs supported: SAE J1979-DA)	U	XX_{16}	TID
#7	TID#6 (Test IDs supported: SAE J1979-DA)	U	XX_{16}	TID
U = User Optional. TID may be included to avoid multiple TID supported request messages.				

To request TIDs supported range from $C1_{16}$ to FF_{16}, another request message with TID#1 = $C0_{16}$ and TID#2 = $E0_{16}$ shall be sent to the vehicle.

8.8.2.2 Request control of on-board device response message definition (report supported TIDs)

ECU(s) shall respond to all supported ranges if requested. A range is defined as a block of 32 TIDs (e.g. range #1: TID 01_{16} - 20_{16}). The ECU shall not respond to unsupported TID ranges unless subsequent ranges have a supported TID(s).

Table 202 — Request control of on-board device response message (report supported TIDs)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request control of on-board device response message SID	M	48_{16}	SIDPR
	data record of supported TIDs = [			TIDREC_
[a] C1 = Conditional. TID value shall be the same value as included in the request message if supported by the ECU.				
[b] C2 = Conditional. Value indicates TIDs supported; range of supported TIDs depends on selected TID value (see C1).				

Table 202 *(continued)*

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#2	1st supported TID	M	XX_{16}	TID
#3	Data A: supported TIDs,	M	XX_{16}	DATA_A
#4	Data B: supported TIDs,	M	XX_{16}	DATA_B
#5	Data C: supported TIDs,	M	XX_{16}	DATA_C
#6	Data D: supported TIDs]	M	XX_{16}	DATA_D
:	:	:	:	:
	data record of supported TIDs = [			TIDREC_
#n-4	mth supported TID	C1[a]	XX_{16}	TID
#n-3	Data A: supported TIDs,	C2[b]	XX_{16}	DATA_A
#n-2	Data B: supported TIDs,	C2	XX_{16}	DATA_B
#n-1	Data C: supported TIDs,	C2	XX_{16}	DATA_C
#n	Data D: supported PIDs]	C2	XX_{16}	DATA_D

[a] C1 = Conditional. TID value shall be the same value as included in the request message if supported by the ECU.

[b] C2 = Conditional. Value indicates TIDs supported; range of supported TIDs depends on selected TID value (see C1).

The response message shall only include the TID(s) and Data A to D which are supported by the ECU. If the request message includes a TID value(s) which is (are) not supported by the ECU, those shall not be included in the response message.

8.8.2.3 Request control of on-board system request message definition (read TID values)

Table 203 — Request control of on-board device request message (read TID values)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request control of on-board device request SID	M	08_{16}	SIDRQ
	data record of Test ID = [			TIDREC_
#2	Test ID (request Test ID values)	M/C1[a]	XX_{16}	TID
#3	Data A,	C2[b]	XX_{16}	DATA_A
#4	Data B,	C2	XX_{16}	DATA_B
#5	Data C,	C2	XX_{16}	DATA_C
#6	Data D,	C2	XX_{16}	DATA_D
#7	Data E]	C2	XX_{16}	DATA_E

[a] C1 = Conditional. Test ID value shall be one of the supported Test IDs of previous response message.

[b] C2 = Conditional. Presence and values of Data A to E parameter depend on Test ID.

8.8.2.4 Request control of on-board device response message definition (report TID values)

Table 204 — Request control of on-board device response message (report TID values)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request control of on-board device response SID	M	48_{16}	SIDPR
	data record of Test ID = [			TIDREC_
#2	Test ID (report Test ID values)	M/C1[a]	XX_{16}	TID
#3	Data A,	C2[b]	XX_{16}	DATA_A
#4	Data B,	C2	XX_{16}	DATA_B
#5	Data C,	C2	XX_{16}	DATA_C
#6	Data D,	C2	XX_{16}	DATA_D
#7	Data E]	C2	XX_{16}	DATA_E

[a] C1 = Conditional. Test ID value shall be the same value as included in the request message.

[b] C2 = Conditional. Presence and values of Data A to E parameter depend on Test ID.

8.8.3 Parameter definition

8.8.3.1 Test IDs supported

Refer to SAE J1979-DA.

8.8.3.2 Test ID description

Refer to SAE J1979-DA.

8.8.4 Message example

Tables 181 and 182 show how "request control of on-board system, test, or component" service shall be implemented.

8.8.4.1 Step #1: Request control of on-board system, test, or component (request for supported Test IDs)

The external test equipment requests all supported Test IDs from the vehicle. Refer to the example of Service 01_{16} for guidance on requesting supported Test IDs (the same concept is used for supported TIDs).

As a result of the supported TID request, the external test equipment creates an internal list of supported PIDs for each ECU. ECU#1 (ECM) supports Test ID 01_{16}. ECU#2 (TCM) does not support any Test IDs and therefore does not send a response message.

8.8.4.2 Step #2: Request control of on-board device (Service 08_{16}, Test ID 01_{16})

The external test equipment sends a "request control of on-board device" message with one (1) supported Test ID 01_{16} to the vehicle.

Table 205 — 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**	**Mnemonic**
#1	Request control of on-board device request SID	08_{16}	SIDRQ

Table 205 *(continued)*

Message Direction:	External test equipment → All ECUs		
Message Type:	Request		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#2	Test ID: 01 - Evaporative system leak test	01_{16}	TID

Table 206 — 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**	**Mnemonic**
#1	Request control of on-board device response SID	48_{16}	SIDPR
#2	Test ID: 01_{16} - Evaporative system leak test	01_{16}	TID

In Table 184, the conditions of the system are not acceptable to run the evaporative system leak test. Therefore, the ECM (ECU#1) responds with a negative response message with NRC 22_{16} - conditionsNotCorrect. The TCM (ECU#2) does not respond because it previously reported that it does not support the evaporative system leak test.

Table 207 — 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**	**Mnemonic**
#1	Request control of on-board device request SID	08_{16}	SIDRQ
#2	Test ID: 01 - Evaporative system leak test	01_{16}	TID

Table 208 — Negative response message

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Negative Response Service Identifier	$7F_{16}$	SIDNR
#2	Request control of on-board device request SID	08_{16}	SIDRQ
#3	Negative Response Code: conditionsNotCorrect	22_{16}	NR_CNC

8.9 Service 09_{16} — Request vehicle information

8.9.1 Functional description

The purpose of this service is to enable the external test equipment to request vehicle-specific vehicle information such as Vehicle Identification Number (VIN) and Calibration IDs. Some of this information may be required by regulations and some should be reported in a standard format if supported by the vehicle manufacturer. INFOTYPEs are defined in SAE J1979-DA.

Not all INFOTYPEs are applicable or supported by all systems. INFOTYPE 00_{16} is a bit-encoded value that indicates for each ECU which INFOTYPEs are supported. INFOTYPE 00_{16} indicates support for INFOTYPEs from 01_{16} to 20_{16}. INFOTYPE 20_{16} indicates support for INFOTYPEs 21_{16} through 40_{16}, etc. This is the same concept for PIDs/TIDs/INFOTYPEs support in Services 01_{16}, 02_{16}, 06_{16}, 08_{16}, 09_{16}. INFOTYPE 00_{16} is required for those ECUs that respond to a corresponding Service 09_{16} request message as specified in SAE J1979-DA.

The request message including supported INFOTYPEs may contain up to six (6) INFOTYPEs. A request message including an INFOTYPE, which reports vehicle information, shall only contain one (1) INFOTYPE. External test equipment shall not request a combination of INFOTYPEs supported and a single INFOTYPE, which reports vehicle information. The ECU shall support requests for up to six (6) supported INFOTYPEs and only one (1) INFOTYPE, which reports vehicle information.

If INFOTYPE 02_{16} (VIN) is indicated as supported, the ECU shall respond within $P2_{max}$ timing even if the VIN is missing or incomplete. For example, a development ECU may respond with FF_{16} characters for VIN because the VIN has not been programmed.

8.9.2 Message data bytes

8.9.2.1 Request vehicle information request message definition (request supported INFOTYPE)

Table 209 — Request vehicle information request message (request supported INFOTYPE)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request vehicle information request SID	M	09_{16}	SIDRQ
#2	INFOTYPE#1 (INFOTYPEs supported: SAE J1979-DA)	M	XX_{16}	INFTYP
#3	INFOTYPE#2 (INFOTYPEs supported: SAE J1979-DA)	U[a]	XX_{16}	INFTYP
#4	INFOTYPE#3 (INFOTYPEs supported: SAE J1979-DA)	U	XX_{16}	INFTYP
#5	INFOTYPE#4 (INFOTYPEs supported: SAE J1979-DA)	U	XX_{16}	INFTYP
#6	INFOTYPE#5 (INFOTYPEs supported: SAE J1979-DA)	U	XX_{16}	INFTYP
#7	INFOTYPE#6 (INFOTYPEs supported: SAE J1979-DA)	U	XX_{16}	INFTYP

[a] U = User Optional. INFOTYPE may be included to avoid multiple INFOTYPE supported request messages.

To request INFOTYPEs supported range from $C1_{16}$ to FF_{16}, another request message with INFOTYPE#1 = $C0_{16}$ and INFOTYPE#2 = $E0_{16}$ shall be sent to the vehicle.

8.9.2.2 Request vehicle information response message definition (report supported INFOTYPE)

ECU(s) shall respond to all supported ranges if requested. A range is defined as a block of 32 INFOTYPEs (e.g. range #1: INFOTYPE 01_{16} to 20_{16}). The ECU shall not respond to unsupported INFOTYPE ranges unless subsequent ranges have a supported INFOTYPE(s).

Table 210 — Request vehicle information response message (report supported INFOTYPE)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request vehicle information response SID	M	49_{16}	SIDPR
	data record of supported INFOTYPEs = [			INFTYPREC
#2	1st supported INFOTYPE	M	XX_{16}	INFTYP
#3	Data A: supported INFOTYPEs,	M	XX_{16}	DATA_A
#4	Data B: supported INFOTYPEs,	M	XX_{16}	DATA_B
#5	Data C: supported INFOTYPEs,	M	XX_{16}	DATA_C
#6	Data D: supported INFOTYPEs]	M	XX_{16}	DATA_D
:	:	:	:	:
	data record of supported INFOTYPEs = [			INFTYPREC

[a] C1 = Conditional. INFOTYPE value shall be the same value as included in the request message if supported by the ECU.

[b] C2 = Conditional. Value indicates INFOTYPEs supported; range of supported INFOTYPEs depends on selected INFOTYPE value (see C1).

Table 210 *(continued)*

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#n-4	m^{th} supported INFOTYPE	C1[a]	XX_{16}	INFTYP
#n-3	Data A: supported INFOTYPEs,	C2[b]	XX_{16}	DATA_A
#n-2	Data B: supported INFOTYPEs,	C2	XX_{16}	DATA_B
#n-1	Data C: supported INFOTYPEs,	C2	XX_{16}	DATA_C
#n	Data D: supported INFOTYPEs]	C2	XX_{16}	DATA_D

[a] C1 = Conditional. INFOTYPE value shall be the same value as included in the request message if supported by the ECU.

[b] C2 = Conditional. Value indicates INFOTYPEs supported; range of supported INFOTYPEs depends on selected INFOTYPE value (see C1).

The response message shall only include the INFOTYPEs and Data A to D which are supported by the ECU. If the request message includes an INFOTYPE value(s), which is (are) not supported by the ECU, those shall not be included in the response message.

8.9.2.3 Request vehicle information request message definition (read INFOTYPE values)

Table 211 — Request vehicle information request message (read INFOTYPE values)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request vehicle information request SID	M	09_{16}	SIDRQ
#2	INFOTYPE (read INFOTYPE values)	M	XX_{16}	INFTYP

8.9.2.4 Request vehicle information response message definition (report INFOTYPE values)

Table 212 — Request vehicle information response message (report INFOTYPE values)

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request vehicle information response SID	M	49_{16}	SIDPR
	data record of INFOTYPE = [			INFTYPREC
#2	INFOTYPE (report INFOTYPE values)	M/C1[a]	XX_{16}	INFTYP
#3	NOfDataItems	M	XX_{16}	NODI
#4	data #1,	M	XX_{16}	DATA_#1
#5	data #2,	C2[b]	XX_{16}	DATA_#2
:	:	C2	XX_{16}	:
#m	data #m]	C2	XX_{16}	DATA_#m

[a] C1 = Conditional. INFOTYPE value shall be the same value as included in the request message.

[b] C2 = Conditional. Data #1 - #m depend on selected INFOTYPE value.

8.9.3 Parameter definition

8.9.3.1 Vehicle information types supported

Refer to SAE J1979-DA.

8.9.3.2 Vehicle information type description

Refer to SAE J1979-DA.

8.9.3.3 Number of data items data byte description

This parameter defines the number of data items included in the response message which are identified and belong to the INFOTYPE reported.

EXAMPLE A request message with the INFOTYPE for CVN may cause the ECU to send a response message that contains multiple CVNs. The number of CVNs is included in the "number of data items" parameter.

8.9.4 Message example

The tables below show how the "request vehicle information" service shall be implemented.

8.9.4.1 Step #1: Request vehicle information (request supported INFOTYPE) from vehicle

The external test equipment requests all supported INFOTYPEs (INFOTYPE#1 = 00_{16}) from the vehicle. ECU#1 (ECM) and ECU#2 (TCM) send a response message with INFOTYPEs supported information for INFOTYPEs 01_{16} - 20_{16}.

Now the external test equipment creates an internal list of supported INFOTYPEs for each ECU. ECU#1 (ECM) supports the following INFOTYPEs: 02_{16}, 04_{16}, 06_{16}, 08_{16}, and $0A_{16}$. ECU#2 (TCM) supports INFOTYPEs 04_{16} and 06_{16}.

8.9.4.2 Step #2: Request INFOTYPEs from vehicle

Now the external test equipment requests the following INFOTYPEs:

— INFOTYPE 02_{16}: VIN = [1G1JC5444R7252367] supported by ECU#1;

— INFOTYPE 04_{16}: Cal. ID#1 = [JMB*36761500] supported by ECU#1;

— INFOTYPE 04_{16}: Cal. ID#2 = [JMB*4787261111] supported by ECU#1;

— INFOTYPE 06_{16}: Cal. CVN#1 = [1791BC82] supported by ECU#1;

— INFOTYPE 06_{16}: Cal. CVN#2 = [16E062BE] supported by ECU#1;

— INFOTYPE 08_{16}: IPT = [04000D09 … 02BF031B] supported by ECU#1 (spark ignition);

— INFOTYPE $0A_{16}$: ECU Name = [ECU – Engine Control] supported by ECU#1;

— INFOTYPE $0B_{16}$: IPT (compression ignition engines) = [NODI, …] supported by ECU#1;

— INFOTYPE $0D_{16}$: ESN = [BRAND 3217486] supported by ECU #1;

— INFOTYPE $0F_{16}$: EROTAN = [DOC-CR-934567] supported by ECU #1;

— INFOTYPE 12_{16}: FEOCNTR = [FEOCNTR_A, FEOCNTR_B] supported by ECU#1;

— INFOTYPE 04_{16}: Cal. ID = [JMA*431299110000] supported by ECU#2;

— INFOTYPE 06_{16}: Cal. CVN = [98123476] supported by ECU#2.

NOTE A compression ignition engine will support INFOTYPE $0B_{16}$ instead of 08_{16} for In-use Performance Tracking (IPT) data.

Table 213 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: 02_{16} - VIN (Vehicle Identification Number)	02_{16}	INFTYP

Table 214 — 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**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: 02_{16} - VIN (Vehicle Information Number)	02_{16}	INFTYP
#3	Number of data items: 01_{16}	01_{16}	NODI
#4	1st ASCII character of VIN: '1'	31_{16}	VIN
#5	2nd ASCII character of VIN: 'G'	47_{16}	VIN
#6	3rd ASCII character of VIN: '1'	31_{16}	VIN
#7	4th ASCII character of VIN: 'J'	$4A_{16}$	VIN
#8	5th ASCII character of VIN: 'C'	43_{16}	VIN
#9	6th ASCII character of VIN: '5'	35_{16}	VIN
#10	7th ASCII character of VIN: '4'	34_{16}	VIN
#11	8th ASCII character of VIN: '4'	34_{16}	VIN
#12	9th ASCII character of VIN: '4'	34_{16}	VIN
#13	10th ASCII character of VIN: 'R'	52_{16}	VIN
#14	11th ASCII character of VIN: '7'	37_{16}	VIN
#15	12th ASCII character of VIN: '2'	32_{16}	VIN
#16	13th ASCII character of VIN: '5'	35_{16}	VIN
#17	14th ASCII character of VIN: '2'	32_{16}	VIN
#18	15th ASCII character of VIN: '3'	33_{16}	VIN
#19	16th ASCII character of VIN: '6'	36_{16}	VIN
#20	17th ASCII character of VIN: '7'	37_{16}	VIN

Now the external test equipment requests the following INFOTYPE:

— INFOTYPE 04_{16}: CALID#1 = [JMB*36761500] and CALID#2 = [JMB*4787261111]; supported by ECU#1.

— INFOTYPE 04_{16}: CALID#1 = [JMA*431299110000]; supported by ECU#2.

Table 215 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: Calibration ID	04_{16}	INFTYP

Table 216 — 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**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Calibration ID	04_{16}	INFTYP
#3	Number of data items: 02	02_{16}	NODI
#4	Data A: 'J'	$4A_{16}$	DATA_A
#5	Data B: 'M'	$4D_{16}$	DATA_B
#6	Data C: 'B'	42_{16}	DATA_C
#7	Data D: '*'	$2A_{16}$	DATA_D
#8	Data E: '3'	33_{16}	DATA_E
#9	Data F: '6'	36_{16}	DATA_F
#10	Data G: '7'	37_{16}	DATA_G
#11	Data H: '6'	36_{16}	DATA_H
#12	Data I: '1'	31_{16}	DATA_I
#13	Data J: '5'	35_{16}	DATA_J
#14	Data K: '0'	30_{16}	DATA_K
#15	Data L: '0'	30_{16}	DATA_L
#16	Data M: Filler byte	00_{16}	DATA_M
#17	Data N: Filler byte	00_{16}	DATA_N
#18	Data O: Filler byte	00_{16}	DATA_O
#19	Data P: Filler byte	00_{16}	DATA_P
#20	Data A: 'J'	$4A_{16}$	DATA_A
#21	Data B: 'M'	$4D_{16}$	DATA_B
#22	Data C: 'B'	42_{16}	DATA_C
#23	Data D: '*'	$2A_{16}$	DATA_D
#24	Data E: '4'	34_{16}	DATA_E
#25	Data F: '7'	37_{16}	DATA_F
#26	Data G: '8'	38_{16}	DATA_G
#27	Data H: '7'	37_{16}	DATA_H
#28	Data I: '2'	32_{16}	DATA_I
#29	Data J: '6'	36_{16}	DATA_J
#30	Data K: '1'	31_{16}	DATA_K
#31	Data L: '1'	31_{16}	DATA_L
#32	Data M: '1'	31_{16}	DATA_M
#33	Data N: '1'	31_{16}	DATA_N
#34	Data O: Filler byte	00_{16}	DATA_O
#35	Data P: Filler byte	00_{16}	DATA_P

Table 217 — Request vehicle information response message

Message Direction:	ECU#2 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Calibration ID	04_{16}	INFTYP
#3	Number of data items: 01	01_{16}	NODI
#4	Data A: 'J'	$4A_{16}$	DATA_A
#5	Data B: 'M'	$4D_{16}$	DATA_B
#6	Data C: 'A'	41_{16}	DATA_C
#7	Data D: '*'	$2A_{16}$	DATA_D
#8	Data E: '4'	34_{16}	DATA_E
#9	Data F: '3'	33_{16}	DATA_F
#10	Data G: '1'	31_{16}	DATA_G
#11	Data H: '2'	32_{16}	DATA_H
#12	Data I: '9'	39_{16}	DATA_I
#13	Data J: '9'	39_{16}	DATA_J
#14	Data K: '1'	31_{16}	DATA_K
#15	Data L: '1'	31_{16}	DATA_L
#16	Data M: '0'	30_{16}	DATA_M
#17	Data N: '0'	30_{16}	DATA_N
#18	Data O: '0'	30_{16}	DATA_O
#19	Data P: '0'	30_{16}	DATA_P

NOTE The same response message with different data byte content will be sent by ECU#2 in this example.

In the following example, the ECUs need more time than $P2_{CAN}$ to calculate the CVN(s). Therefore, both ECUs respond with negative response messages with NRC 78_{16} - RequestCorrectlyReceived-ResponsePending as long as the positive response message is not ready in the ECU.

Now the external test equipment requests the following INFOTYPE:

— INFOTYPE 06_{16}: CVN#1 = [17_{16} 91_{16} BC_{16} 82_{16}] and CVN#2 = [16_{16} $E0_{16}$ 62_{16} BE_{16}]; supported by ECU#1;

— INFOTYPE 06_{16}: CVN = [98_{16} 12_{16} 34_{16} 76_{16}]; supported by ECU#2.

Table 218 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: Calibration Verification Number	06_{16}	INFTYP

Table 219 — Negative response message

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Negative Response Service Identifier	$7F_{16}$	SIDNR
#2	Request vehicle information request SID	09_{16}	SIDRQ
#3	Negative Response Code: RequestCorrectlyReceived-ResponsePending	78_{16}	NR_RCR_RP

Table 220 — Negative response message

Message Direction:	ECU#2 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Negative Response Service Identifier	$7F_{16}$	SIDNR
#2	Request vehicle information request SID	09_{16}	SIDRQ
#3	Negative Response Code: RequestCorrectlyReceived-ResponsePending	78_{16}	NR_RCR_RP

Table 221 — Request vehicle information response message (1st)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Calibration Verification Number	06_{16}	INFTYP
#3	Number of data items: 02	02_{16}	NODI
#4	Data A: 17_{16}	17_{16}	DATA_A
#5	Data B: 91_{16}	91_{16}	DATA_B
#6	Data C: BC_{16}	BC_{16}	DATA_C
#7	Data D: 82_{16}	82_{16}	DATA_D
#8	Data E: 16_{16}	16_{16}	DATA_E
#9	Data F: $E0_{16}$	$E0_{16}$	DATA_F
#10	Data G: 62_{16}	62_{16}	DATA_G
#11	Data H: BE_{16}	BE_{16}	DATA_H

Table 222 — Request vehicle information response message (1st)

Message Direction:	ECU#2 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: Calibration Verification Number	06_{16}	INFTYP
#3	Number of data items: 01	01_{16}	NODI
#4	Data A: 98_{16}	98_{16}	DATA_A
#5	Data B: 12_{16}	12_{16}	DATA_B

Table 222 *(continued)*

Message Direction:	ECU#2 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#6	Data C: 34_{16}	34_{16}	DATA_C
#7	Data D: 76_{16}	76_{16}	DATA_D

Now, for a spark ignition engine, the external test equipment requests the following INFOTYPE:

— INFOTYPE 08_{16}: IPT; supported by ECU#1.

Table 223 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: In-use Performance Tracking	08_{16}	INFTYP

Table 224 — Request vehicle information response message (1)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: In-use Performance Tracking	08_{16}	INFTYP
#3	Number of data items: 20 (some vehicles will report 16 items)	14_{16}	NODI
#4	OBDCOND_A: 1 024 counts	04_{16}	OBDCOND_A
#5	OBDCOND_B: 1 024 counts	00_{16}	OBDCOND_B
#6	IGNCNTR_A: 3 337 counts	$0D_{16}$	IGNCNTR_A
#7	IGNCNTR_B: 3 337 counts	09_{16}	IGNCNTR_B
#8	CATCOMP1_A: 824 counts	03_{16}	CATCOMP1_A
#9	CATCOMP1_B: 824 counts	38_{16}	CATCOMP1_B
#10	CATCOND1_A: 945 counts	03_{16}	CATCOND1_A
#11	CATCOND1_B: 945 counts	$B1_{16}$	CATCOND1_B
#12	CATCOMP2_A: 711 counts	02_{16}	CATCOMP2_A
#13	CATCOMP2_B: 711 counts	$C7_{16}$	CATCOMP2_B
#14	CATCOND2_A: 945 counts	03_{16}	CATCOND2_A
#15	CATCOND2_B: 945 counts	$B1_{16}$	CATCOND2_B
#16	O2SCOMP1_A: 737 counts	02_{16}	O2SCOMP1_A
#17	O2SCOMP1_B: 737 counts	$E1_{16}$	O2SCOMP1_B
#18	O2SCOND1_A: 924 counts	03_{16}	O2SCOND1_A
#19	O2SCOND1_B: 924 counts	$9C_{16}$	O2SCOND1_B
#20	O2SCOMP2_A: 724 counts	02_{16}	O2SCOMP2_A
#21	O2SCOMP2_B: 724 counts	$D4_{16}$	O2SCOMP2_B
#22	O2SCOND2_A: 833 counts	03_{16}	O2SCOND2_A

Table 224 *(continued)*

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#23	O2SCOND2_B: 833 counts	41_{16}	O2SCOND2_B
#24	EGRCOMP_A: 997 counts	03_{16}	EGRCOMP_A
#25	EGRCOMP_B: 997 counts	$E5_{16}$	EGRCOMP_B
#26	EGRCOND_A: 1 010 counts	03_{16}	EGRCOND_A
#27	EGRCOND_B: 1 010 counts	$F2_{16}$	EGRCOND_B
#28	AIRCOMP_A: 937 counts	03_{16}	AIRCOMP_A
#29	AIRCOMP_B: 937 counts	$A9_{16}$	AIRCOMP_B
#30	AIRCOND_A: 973 counts	03_{16}	AIRCOND_A
#31	AIRCOND_B: 973 counts	CD_{16}	AIRCOND_B
#32	EVAPCOMP_A: 68 counts	00_{16}	EVAPCOMP_A
#33	EVAPCOMP_B: 68 counts	44_{16}	EVAPCOMP_B
#34	EVAPCOND_A: 97 counts	00_{16}	EVAPCOND_A
#35	EVAPCOND_B: 97 counts	61_{16}	EVAPCOND_B
#36	SO2SCOMP1_A 677 counts	02_{16}	SO2SCOMP1_A
#37	SO2SCOMP1_B: 677 counts	$A5_{16}$	SO2SCOMP1_B
#38	SO2SCOND1_A: 824 counts	03_{16}	SO2SCOND1_A
#39	SO2SCOND1_B: 824 counts	38_{16}	SO2SCOND1_B
#40	SO2SCOMP2_A: 703 counts	02_{16}	SO2SCOMP2_A
#41	SO2SCOMP2_B: 703 counts	BF_{16}	SO2SCOMP2_B
#42	SO2SCOND2_A: 795 counts	03_{16}	SO2SCOND2_A
#43	SO2SCOND2_B: 795 counts	$1B_{16}$	SO2SCOND2_B

Now the external test equipment requests the following INFOTYPE:

— INFOTYPE $0A_{16}$: ECUNAME; supported by ECU#1; The name of the ECU is: “ECM-EngineControl”.

Table 225 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: ECU’s/module’s acronym and text name	$0A_{16}$	INFTYP

Table 226 — Request vehicle information response message (1)

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: ECU’s/module’s acronym and text name	$0A_{16}$	INFTYP
#3	Number of data items: 1	01_{16}	NODI

Table 226 *(continued)*

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#4	Data A: 'E'	45_{16}	ECUNAME_A
#5	Data B: 'C'	43_{16}	ECUNAME_B
#6	Data C: 'M'	$4D_{16}$	ECUNAME_C
#7	Data D: '1' (or filler byte, 00_{16}, if single ECM in the vehicle)	31_{16}	ECUNAME_D
#8	Data E: '-' delimiter	$2D_{16}$	ECUNAME_E
#9	Data F: 'E'	45_{16}	ECUNAME_F
#10	Data G: 'n'	$6E_{16}$	ECUNAME_G
#11	Data H: 'g'	67_{16}	ECUNAME_H
#12	Data I: 'i'	69_{16}	ECUNAME_I
#13	Data J: 'n'	$6E_{16}$	ECUNAME_J
#14	Data K: 'e'	65_{16}	ECUNAME_K
#15	Data L: 'C'	43_{16}	ECUNAME_L
#16	Data M: 'o'	$6F_{16}$	ECUNAME_M
#17	Data N: 'n'	$6E_{16}$	ECUNAME_N
#18	Data O: 't'	74_{16}	ECUNAME_O
#19	Data P: 'r'	72_{16}	ECUNAME_P
#20	Data Q: 'o'	$6F_{16}$	ECUNAME_Q
#21	Data R: 'l'	$6C_{16}$	ECUNAME_R
#22	Data S: filler byte	00_{16}	ECUNAME_S
#23	Data T: filler byte	00_{16}	ECUNAME_T

The external test equipment requests the following INFOTYPE for a compression ignition engine:

— INFOTYPE $0B_{16}$: IPT for compression ignition engines; supported by ECU#1.

Table 227 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: In-use Performance Tracking	$0B_{16}$	INFTYP

Table 228 — 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**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: In-use Performance Tracking	$0B_{16}$	INFTYP
#3	Number of data items: 16 (some vehicles will report 18 items)	10_{16}	NODI
#4	OBDCOND_A: 1 024 counts	04_{16}	OBDCOND_A

Table 228 *(continued)*

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#5	OBDCOND_B: 1 024 counts	00_{16}	OBDCOND_B
#6	IGNCNTR_A: 3 337 counts	$0D_{16}$	IGNCNTR_A
#7	IGNCNTR_B: 3 337 counts	09_{16}	IGNCNTR_B
#8	HCCATCOMP_A: 824 counts	03_{16}	HCCATCOMP_A
#9	HCCATCOMP_B: 824 counts	38_{16}	HCCATCOMP_B
#10	HCCATCOND_A: 945 counts	03_{16}	HCCATCOND_A
#11	HCCATCOND_B: 945 counts	$B1_{16}$	HCCATCOND_B
#12	NCATCOMP_A: 711 counts	02_{16}	NCATCOMP_A
#13	NCATCOMP_B: 711 counts	$C7_{16}$	NCATCOMP_B
#14	NCATCOND_A: 945 counts	03_{16}	NCATCOND_A
#15	NCATCOND_B: 945 counts	$B1_{16}$	NCATCOND_B
#16	NADSCOMP_A: 737 counts	02_{16}	NADSCOMP_A
#17	NADSCOMP_B: 737 counts	$E1_{16}$	NADSCOMP_B
#18	NADSCOND_A: 924 counts	03_{16}	NADSCOND_A
#19	NADSCOND_B: 924 counts	$9C_{16}$	NADSCOND_B
#20	PMCOMP_A: 724 counts	02_{16}	PMCOMP_A
#21	PMCOMP_B: 724 counts	$D4_{16}$	PMCOMP_B
#22	PMCOND_A: 833 counts	03_{16}	PMCOND_A
#23	PMCOND_B: 833 counts	41_{16}	PMCOND_B
#24	EGSCOMP_A: 997 counts	03_{16}	EGSCOMP_A
#25	EGSCOMP_B: 997 counts	$E5_{16}$	EGSCOMP_B
#26	EGSCOND_A: 1 010 counts	03_{16}	EGSCOND_A
#27	EGSCOND_B: 1 010 counts	$F2_{16}$	EGSCOND_B
#28	EGRCOMP_A: 937 counts	03_{16}	EGRCOMP_A
#29	EGRCOMP_B: 937 counts	$A9_{16}$	EGRCOMP_B
#30	EGRCOND_A: 973 counts	03_{16}	EGRCOND_A
#31	EGRCOND_B: 973 counts	CD_{16}	EGRCOND_B
#32	BPCOMP_A: 68 counts	00_{16}	BPCOMP_A
#33	BPCOMP_B: 68 counts	44_{16}	BPCOMP_B
#34	BPCOND_A: 97 counts	00_{16}	BPCOND_A
#35	BPCOND_B: 97 counts	61_{16}	BPCOND_B

Now the external test equipment requests the following InfoType:

— INFOTYPE $0D_{16}$: EngineSerialNumber; supported by ECU#1; The name of the ECU is: “BRAND 3217486”

Table 229 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	INFOTYPE: ESN (Engine Serial Number)	$0D_{16}$	INFTYP

Table 230 — Request vehicle information response message (1)

Message Direction:	ECU #1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: $0D_{16}$ - ESN (Engine Serial Number)	$0D_{16}$	INFTYP
#3	Number of data items: 1	01_{16}	NODI
#4	filler byte	00_{16}	ESN
#5	filler byte	00_{16}	ESN
#6	filler byte	00_{16}	ESN
#7	filler byte	00_{16}	ESN
#8	5th ASCII character of ESN: ‘B’	42_{16}	ESN
#9	6th ASCII character of ESN: ‘R’	52_{16}	ESN
#10	7th ASCII character of ESN: ‘A’	41_{16}	ESN
#11	8th ASCII character of ESN: ‘N’	$4E_{16}$	ESN
#12	9th ASCII character of ESN: ‘D’	44_{16}	ESN
#13	10th ASCII character of ESN: ‘ ’	20_{16}	ESN
#14	11th ASCII character of ESN: ‘3’	33_{16}	ESN
#15	12th ASCII character of ESN: ‘2’	32_{16}	ESN
#16	13th ASCII character of ESN: ‘1’	31_{16}	ESN
#17	14th ASCII character of ESN: ‘7’	37_{16}	ESN
#18	15th ASCII character of ESN: ‘4’	34_{16}	ESN
#19	16th ASCII character of ESN: ‘8’	38_{16}	ESN
#20	17th ASCII character of ESN: ‘6’	36_{16}	ESN

Now the external test equipment requests the following InfoType:

— INFOTYPE $0F_{16}$: Exhaust Regulation Or Type Approval Number; supported by ECU#1; The name of the ECU is: “DOC-CR-934567”

Table 231 — 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**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	InfoType: Exhaust Regulation Or Type Approval Number	$0F_{16}$	INFTYP

Table 232 — Request vehicle information response message (1)

Message Direction:	ECU #1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	INFOTYPE: $0F_{16}$ - EROTAN	$0F_{16}$	INFTYP
#3	Number of data items: 1	01_{16}	NODI
#4	filler byte	00_{16}	EROTAN
#5	filler byte	00_{16}	EROTAN
#6	filler byte	00_{16}	EROTAN
#7	filler byte	00_{16}	EROTAN
#8	5th ASCII character of EROTAN: 'D'	44_{16}	EROTAN
#9	6th ASCII character of EROTAN: 'O'	$4F_{16}$	EROTAN
#10	7th ASCII character of EROTAN: 'C'	43_{16}	EROTAN
#11	8th ASCII character of EROTAN: '-'	$2D_{16}$	EROTAN
#12	9th ASCII character of EROTAN: 'C'	43_{16}	EROTAN
#13	10th ASCII character of EROTAN: 'R'	52_{16}	EROTAN
#14	11th ASCII character of EROTAN: '-'	$2D_{16}$	EROTAN
#15	12th ASCII character of EROTAN: '9'	39_{16}	EROTAN
#16	13th ASCII character of EROTAN: '3'	33_{16}	EROTAN
#17	14th ASCII character of EROTAN: '4'	34_{16}	EROTAN
#18	15th ASCII character of EROTAN: '5'	35_{16}	EROTAN
#19	16th ASCII character of EROTAN: '6'	36_{16}	EROTAN
#20	17th ASCII character of EROTAN: '7'	37_{16}	EROTAN

Now the external test equipment requests the following InfoType:

— InfoType $12: Fueled Engine Operation Ignition Counter; supported by ECU#1

Table 233 — Request vehicle information request message

Message Direction:	External test equipment → All ECUs		
Messsage Type:	Request		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#1	Request vehicle information request SID	09_{16}	SIDRQ
#2	InfoType: FEOCNTR	12_{16}	INFTYP

Table 234 — 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**	**Mnemonic**
#1	Request vehicle information response SID	49_{16}	SIDPR
#2	InfoType: Calibration ID	12_{16}	INFTYP
#3	Number of data items: 1	01_{16}	NODI
#4	FEOCNTR_A: 2 390 counts	09_{16}	FEOCNTR_A

Table 234 *(continued)*

Message Direction:	ECU#1 → External test equipment		
Message Type:	Response		
Data Byte	**Description (all values are in hexadecimal)**	**Byte Value**	**Mnemonic**
#5	FEOCNTR_B: 2 390 counts	56_{16}	FEOCNTR_B

8.10 Service $0A_{16}$ — Request emission-related diagnostic trouble codes with permanent status

8.10.1 Functional description

The purpose of this service is to enable the external test equipment to obtain all DTCs with "permanent DTC status". These are DTCs that are "confirmed" and are retained in the non-volatile memory of the server until the appropriate monitor for each DTC has determined that the malfunction is no longer present and is not commanding the MIL on.

Service $0A_{16}$ is required for all emissions-related DTCs. The intended use of this data is to prevent vehicles from passing an in-use inspection simply by disconnecting the battery or clearing DTCs with a scan tool prior to the inspection. The presence of permanent DTCs at an inspection without the MIL illuminated is an indication that a proper repair was not verified by the on-board monitoring system.

Permanent DTCs shall be stored in non-volatile memory (NVRAM) and may not be erased by any diagnostic services (generic or enhanced) or by disconnecting power to the ECU.

A confirmed DTC shall be stored as a permanent DTC no later than the end of the ignition cycle and subsequently at all times that the confirmed DTC is commanding the MIL on (e.g. for currently failing systems but not during the 40 warm-up cycle self-healing process).

Permanent DTCs may be erased if:

— The OBD system itself determines that the malfunction that caused the permanent fault code to be stored is no longer present and is not commanding the MIL on, e.g. three consecutive complete driving cycles with no malfunction, or as specified by the OBD regulations;

— After clearing fault information in the ECU (i.e. through the use of a diagnostic service or battery disconnect):

  — For monitors subject to minimum in-use ratio requirement, the diagnostic monitor for the malfunction that caused the permanent DTC to be stored has fully executed (i.e. has executed the minimum number of checks necessary for MIL illumination) and determined the malfunction is no longer present, e.g. one complete driving cycle with no malfunction or as specified by the OBD regulations;

  — For monitors not subject to minimum in-use ratio requirement, the diagnostic monitor for the malfunction that caused the permanent DTC to be stored has fully executed (i.e. has executed the minimum number of checks necessary for MIL illumination) and determined the malfunction is no longer present, e.g. one complete driving cycle with no malfunction or as specified by the OBD regulations, and the vehicle has completed a standard driving cycle used to increment the in-use general denominator.

— Permanent fault codes may be erased when the ECU containing the permanent DTCs is reprogrammed if the readiness status for all monitored components and systems is set to "not complete" in conjunction with the reprogramming event.

NOTE Due to implementation timing differences during the phase-in of permanent DTCs, there may be cases where some ECUs support permanent DTCs while other ECUs do not within the same vehicle.

8.10.2 Message data bytes

8.10.2.1 Request emission-related diagnostic trouble codes with permanent status request message

Table 235 — Request emission-related diagnostic trouble codes with permanent status request message

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request emission-related diagnostic trouble codes with permanent status request SID	M	$0A_{16}$	SIDRQ

8.10.2.2 Request emission-related diagnostic trouble codes with permanent status response message definition

Table 236 — Request emission-related diagnostic trouble codes with permanent status response message

Data Byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	Request emission-related diagnostic trouble codes with permanent status response SID	M	$4A_{16}$	SIDPR
#2	# of DTC = [	M		#OFDTC
	no emission-related DTCs with permanent status		00_{16}	
	# of emission-related DTCs with permanent status]		01_{16} – FF_{16}	
#3	DTC#1 (High Byte)	C[a]	XX_{16}	DTC1HI
#4	DTC#1 (Low Byte)	C	XX_{16}	DTC1LO
:	:	:	XX_{16}	
#n-1	DTC#m (High Byte)	C	XX_{16}	DTCmHI
#n	DTC#m (Low Byte)	C	XX_{16}	DTCmLO

[a] C = Conditional. DTC#1 - DTC#m are only included if # of DTC parameter value ≠ 00_{16}.

8.10.3 Parameter definition

The # of DTC parameter reports the emission-related DTC(s) currently (at the time of the request message processing) stored in the ECU(s).

8.10.4 Message example

Refer to message example of Service 03_{16}.

Bibliography

[1] ISO 11898-1, *Road vehicles Controller area network (CAN) — Part 1: Data link layer and physical signalling*

[2] ISO 11898-2, *Road vehicles — Controller area network (CAN) — Part 2: High-speed medium access unit*

[3] ISO 14230-1, *Road vehicles — Diagnostic communication over K-Line (DoK-Line) — Part 1: Physical layer*

[4] ISO 15031-3, *Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics — Part 3: Diagnostic connector and related electrical circuits, specification and use*

[5] ISO 15031-1, *Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics — Part 1: General information and use case definition*

[6] ISO 15031-4, *Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics — Part 4: External test equipmen*t

[7] ISO 15031-6, *Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics — Part 6: Diagnostic trouble code definitions*

[8] ISO/IEC 7498-1, *Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model — Part 1*

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

[10] SAE J1850, *Class B Data Communications Network Interface*

[11] SAE J1978, *OBD II Scan Tool*

ICS 13.040.50; 43.040.10

Price based on 140 pages