INTERNATIONAL STANDARD

ISO 27145-3

First edition
2012-08-15

Road vehicles — Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements —

Part 3: **Common message dictionary**

Véhicules routiers — Mise en application des exigences de communication pour le diagnostic embarqué harmonisé à l'échelle mondiale (WWH-OBD) —

Partie 3: Dictionnaire de messages communs

Reference number
ISO 27145-3:2012(E)

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

Published in Switzerland

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.

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

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

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

ISO 27145-3 was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 3, *Electrical and electronic equipment*.

This first edition of ISO 27145-3 cancels and replaces ISO/PAS 27145-3:2006, which has been technically revised.

ISO 27145 consists of the following parts, under the general title *Road vehicles — Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements*:

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

— *Part 2: Common data dictionary*

— *Part 3: Common message dictionary*

— *Part 4: Connection between vehicle and test equipment*

The following parts are under preparation:

— *Part 6: External test equipment*

0 Introduction

0.1 Overview

The ISO 27145 series includes the communication between the vehicle's on-board diagnostics (OBD) systems and external test equipment within the scope of the World-Wide Harmonized On-Board Diagnostics Global Technical Regulations (WWH-OBD GTR).

It has been established in order to apply the unified diagnostic services (specified in ISO 14229-1) to WWH-OBD systems.

The ISO 27145 series includes the communication between the vehicle's WWH-OBD systems and external (off-board) "generic" test equipment within the scope of the country-specific regulatory requirements.

To achieve this, it is based on the Open Systems Interconnection (OSI) Basic Reference Model specified in ISO/IEC 7498-1 and ISO/IEC 10731, which structures communication systems into seven layers. When mapped on this model, the services specified by ISO 27145 are divided into

— diagnostic services (layer 7), specified in ISO 27145-3 with reference to ISO 14229-1,

— presentation layer (layer 6), specified in ISO 27145-2 with reference to SAE J1930-DA, SAE J1939 Companion Spreadsheet (SPNs), SAE J1939-73:2010, Appendix A (FMIs), SAE J1979-DA and SAE J2012-DA,

— session layer services (layer 5), specified in ISO 14229-2,

— transport layer services (layer 4), specified in ISO 27145-4 with reference to ISO 13400-2, ISO 15765-2 and ISO 15765-4,

— network layer services (layer 3), specified in ISO 27145-4 with reference to ISO 15765-4, ISO 15765-2 and ISO 13400-2,

— data link layer (layer 2), specified in ISO 27145-4 with reference to ISO 11898-1, ISO 11898-2, ISO 15765-4, ISO 13400-3 and IEEE 802.3, and

— physical layer (layer 1), specified in ISO 27145-4 with reference to ISO 11898-1, ISO 11898-2, ISO 15765-4, ISO 13400-3 and IEEE 802.3,

in accordance with Table 1.

Table 1 — WWH-OBD specification reference applicable to the OSI layers

Applicability	OSI seven layer	WWH-OBD document reference		
Seven layers according to ISO/IEC 7498-1 and ISO/IEC 10731	Application (layer 7)	ISO 14229-1, ISO 27145-3		
	Presentation (layer 6)	ISO 27145-2, SAE J1930-DA, SAE J1939 Companion Spreadsheet (SPNs), SAE J1939-73:2010, Appendix A (FMIs), SAE J1979-DA, SAE J2012-DA		
	Session (layer 5)	ISO 14229-2		
	Transport (layer 4)	ISO 15765-2 DoCAN, ISO 15765-4 DoCAN	ISO 27145-4	ISO 13400-2 DoIP TCP and IP
	Network (layer 3)			
	Data link (layer 2)	ISO 11898-1 CAN DLL, ISO 11898-2 CAN HS, ISO 15765-4 DoCAN		ISO 13400-3 DoIP, IEEE 802.3
	Physical (layer 1)			

0.2 SAE document reference concept

ISO 27145 makes reference to several SAE documents which contain the terms, data and diagnostic trouble code (DTC) definitions.

ISO 27145-2 defines a common data dictionary for the ISO 27145 series, according to the definitions in the following documents (see Figure 1).

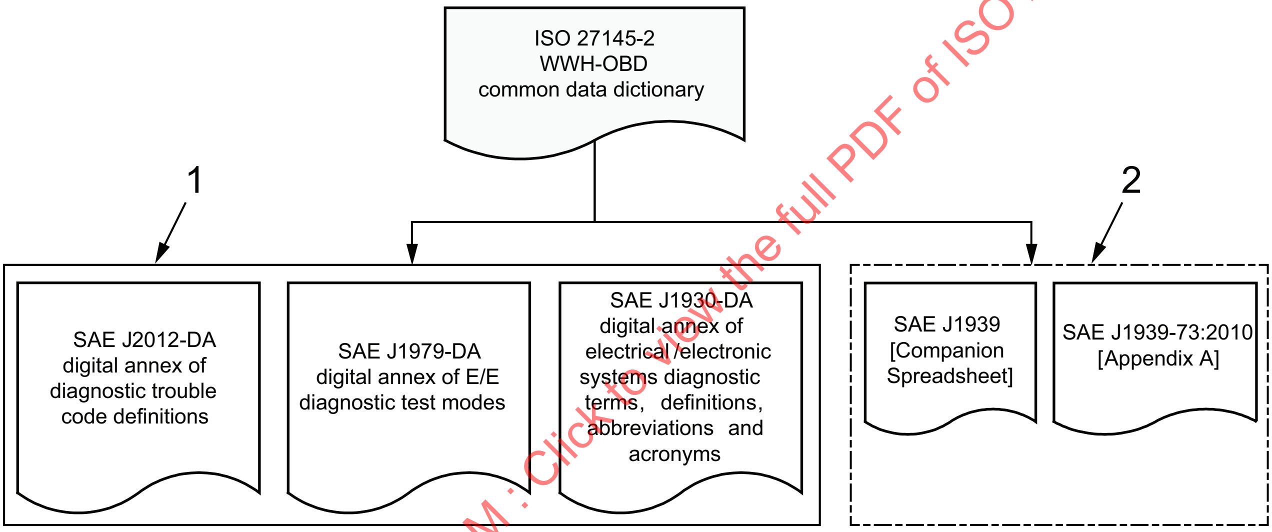
— SAE J1930-DA: this digital annex contains all standardized naming objects, terms and abbreviated terms.

— SAE J1939 Companion Spreadsheet and SAE J1939-73: SAE J1939 Companion Spreadsheet indexes names for suspect parameter numbers (SPNs) that provide an alternative presentation format for SAE J2012-DA DTCs. SPNs are combined with failure mode indicators (FMIs) to form the full alternative presentation. FMIs are described in SAE J1939-73:2010, Appendix A.

NOTE The SAE J1939 Companion Spreadsheet is a document which supplements the SAE J1939 family of standards and contains SPNs and parameter group numbers (PGNs).

— SAE J1979-DA: this digital annex contains all standardized data items such as data identifiers (DIDs), test identifiers (TIDs), monitor identifiers (MIDs) and infotype identifiers (ITIDs).

— SAE J2012-DA: this digital annex contains all standardized data items such as DTC definitions and FTB (failure type byte) definitions.



Key

1 SAE digital annexes: data definitions

2 SAE J1939 series of documents: DTC definitions

Figure 1 — SAE digital annex document reference

0.3 SAE digital annex revision procedure

New regulatory requirements drive new in-vehicle technology to lower emissions, improve safety, etc. It is important to standardize new technology-related OBD monitor data and DTCs in order 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.

ISO 27145-2 references a "Change request form" for use with new data items to be defined by the SAE task force for standardization. It is intended that the standardized data items be defined in SAE J1930-DA, SAE J1979-DA, SAE J2012-DA and SAE J1939. It is intended that the documents be published on the SAE store website once the information has been balloted and approved.

The revision request forms and instructions for updating the registers to ISO 27145 can be obtained on the following data registration websites:

— For SAE J1930-DA: http://www.sae.org/servlets/works/committeeHome.do?comtID=TEVDS7

The column entitled "Resources" shows a document with the title: J1930-DA_Revision_Request_Form.doc. Double click on the name to download the document with the filename: "SAE_J1930-DA_Revision_Request_Form.doc".

— For SAE J1939: http://www.sae.org/

Search "J1939 Request", select "J1939 Request Processing Group", and select "J1939 Request Processing Form and Guidelines".

— For SAE J1979-DA: http://www.sae.org/servlets/works/committeeHome.do?comtID=TEVDS14

The column entitled "Resources" shows a document with the title: J1979-DA_Revision_Request_Form.doc. Double click on the name to download the document with the filename: "SAE_J1979-DA_Revision_Request_Form.doc".

— For SAE J2012-DA: http://www.sae.org/servlets/works/committeeHome.do?comtID=TEVDS9

The column entitled "Resources" shows a document with the title: J2012-DA_Revision_Request_Form.doc. Double click on the name to download the document with the filename: "SAE_J2012-DA_Revision_Request_Form.doc".

It is intended that the revision request form be filled out with the request.

It is intended that e-mails with completed revision request forms as attachments be sent to:

E-mail: saej1930@sae.org

E-mail: saej1979@sae.org

E-mail: saej2012@sae.org

E-mail: saej1939@sae.org

Road vehicles — Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements —

Part 3: Common message dictionary

1 Scope

This part of ISO 27145 defines the implementation of a subset of unified diagnostic services (UDS) specified in ISO 14229-1. The diagnostic services are used to communicate the diagnostic data defined in ISO 27145-2.

The subset of unified diagnostic services derives from the requirements stated in the WWH-OBD GTR (Global technical regulation No. 5; see Reference [17]). The common message set defined in this part of ISO 27145 is independent of the underlying transport, network, data link and physical layer. This part of ISO 27145 does not specify any requirements for the in-vehicle network architecture.

This part of ISO 27145 is compatible with ISO 14229-1 and includes provisions to support the data set of SAE J1979-DA and SAE J2012-DA WWH-OBD.

This part of ISO 27145 is intended for use with ISO 27145-4, which is the entry point for the protocol initialization and is based on two different data links:

— Diagnostic communication over Controller Area Network (DoCAN), ISO 15765-1, ISO 15765-2, ISO 15765-4;

— Diagnostic communication over Internet Protocol (DoIP), ISO 13400 (all parts).

Due to the usage of standard network layer protocols, future extensions to optional physical layers (e.g. wireless) are possible.

Based on the results of the initialization, the external test equipment determines which protocol and diagnostic services are supported by the vehicle's emissions-related system, i.e.

— legislated OBD: ISO 15031 (all parts);

— legislated WWH-OBD: ISO 27145 (all parts).

This part of ISO 27145 includes capabilities required to satisfy OBD regulations for multiple regions, vehicle types, model years, and engine types. Those regulations are not yet final for some regions and are expected to change in the future. This part of ISO 27145 does not attempt to interpret the regulations and does not include applicability of the included diagnostic services and data parameters for various vehicle applications. It is intended that users of this part of ISO 27145 verify the applicability of each of its clauses for a specific vehicle, engine, model year and region.

2 Normative references

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

ISO 14229-1, *Road vehicles — Unified diagnostic services (UDS) — Part 1: Specification and requirements*

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

ISO 27145-1, *Road vehicles — Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements — Part 1: General information and use case definition*

ISO 27145-2, *Road vehicles — Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements — Part 2: Common data dictionary (CDD)*

ISO 27145-4, *Road vehicles — Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements — Part 4: Connection between vehicle and test equipment*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 27145-1, ISO 27145-2, ISO 14229-1 and the following apply.

3.1.1
global technical regulation
GTR
agreement establishing global technical regulations for wheeled vehicles and equipment and parts that can be fitted and/or used on wheeled vehicles

3.2 Abbreviated terms

ASCII	American standard code for information interchange
CDTC	confirmed DTC
DID	diagnostic data identifier
DoCAN	diagnostic communication over controller area network
DoIP	diagnostic communication over internet protocol
DTC	diagnostic trouble code
DTCHB	DTC high byte
DTCLB	DTC low byte
DTCMB	DTC middle byte
DTCS	DTC severity
ECM	engine control module
ECU	electronic control unit
FMI	failure mode indicator
FTB	failure type byte
GTR	global technical regulations
MI	malfunction indicator
N/A	not applicable
PDTC	pending DTC
PDU	protocol data unit

RID	routine identifier
SFID	sub-function identifier
SID	service identifier
SODTC	status of DTC
SPN	suspect parameter number
TNCSLC	test not completed since last clear
TNCTOC	test not completed this operation cycle
UDS	unified diagnostic services
VIN	vehicle identification number
WUC	warm-up cycle
WWH-OBD	world-wide harmonized on-board diagnostics

4 Conventions

The ISO 27145 series is based on the conventions discussed in the OSI Service Conventions (ISO/IEC 10731) as they apply to diagnostic services.

5 Document overview

Figure 2 shows the reference documents for the ISO 27145 series.

The ISO 27145 series specifies or includes the following references:

a) ISO 27145-1 specifies the general structure of the ISO 27145 series and the use cases applicable to WWH-OBD GTR.

b) ISO 27145-2 specifies the common data dictionary with references to:

   1) SAE J1930-DA, which defines the terms, definitions, abbreviated terms, etc.;

   2) SAE J1939 Companion Spreadsheet, which specifies the SPNs;

   3) SAE J1939-73:2010, Appendix A, which specifies the FMIs;

   4) SAE 1979-DA, which specifies all data items;

   5) SAE J2012-DA, which specifies the DTC definitions and FTB definitions.

NOTE The SAE J1939 series of documents is concerned with the definition of emissions-related SPNs and FMIs for use as DTCs.

c) This part of ISO 27145 specifies the diagnostic services defined in ISO 14229-1 that are applicable to WWH-OBD GTR.

d) ISO 14229-2 specifies the standardized service primitive interface to separate application and session layers from protocol transport and network layers.

e) ISO 27145-4 specifies the initialization procedure and includes references to:

   1) ISO 15765-4 DoCAN;

   2) ISO 13400 (all parts) DoIP.

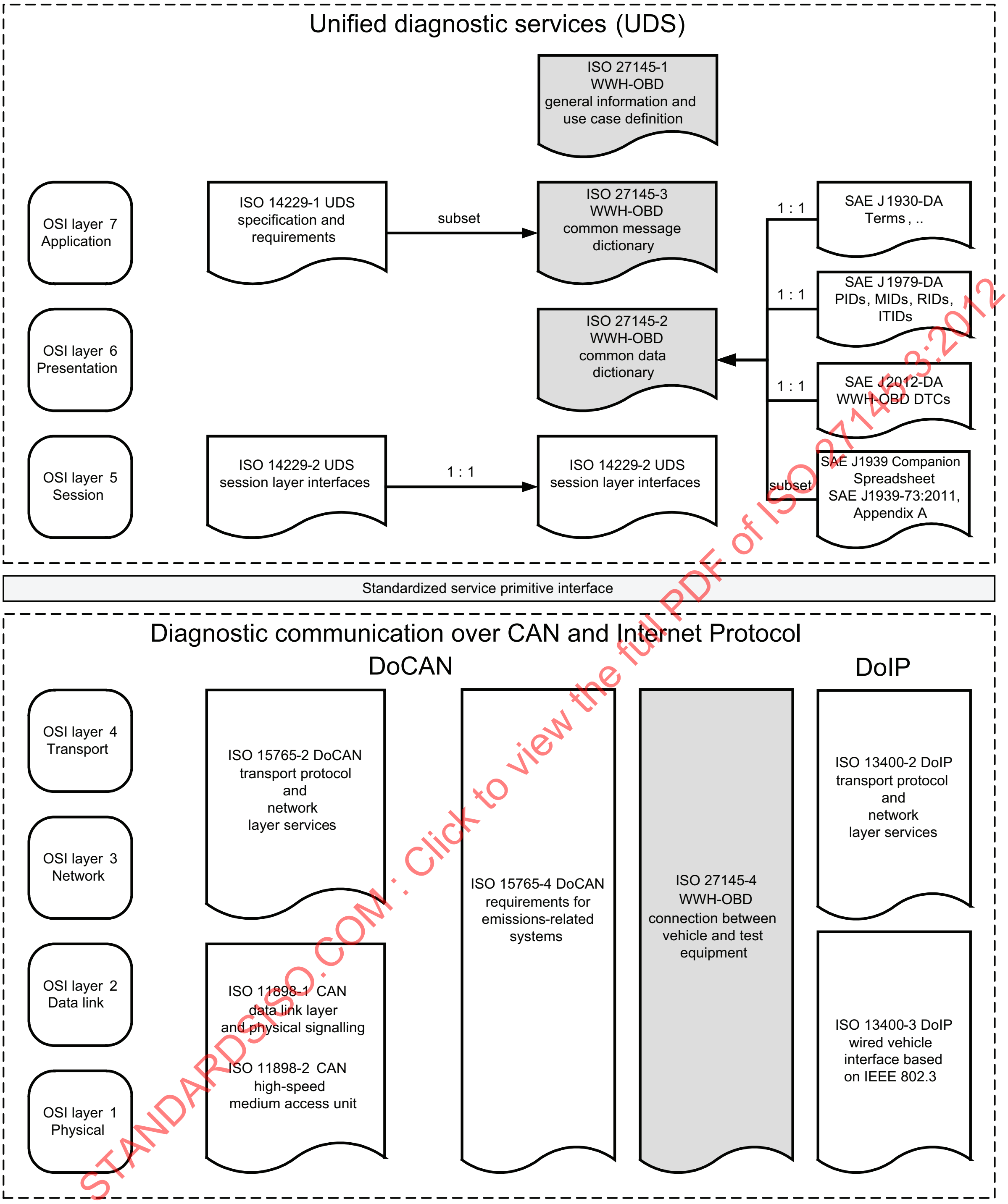

Unified diagnostic services (UDS)
ISO 27145-1 WWH-OBD general information and use case definition
OSI layer 7 Application
ISO 14229-1 UDS specification and requirements
subset
ISO 27145-3 WWH-OBD common message dictionary
1 : 1
SAE J1930-DA Terms, ..
1 : 1
SAE J1979-DA PIDs, MIDs, RIDs, ITIDs
OSI layer 6 Presentation
ISO 27145-2 WWH-OBD common data dictionary
1 : 1
SAE J2012-DA WWH-OBD DTCs
OSI layer 5 Session
ISO 14229-2 UDS session layer interfaces
1 : 1
ISO 14229-2 UDS session layer interfaces
subset
SAE J1939 Companion Spreadsheet SAE J1939-73:2011, Appendix A
Standardized service primitive interface
Diagnostic communication over CAN and Internet Protocol
DoCAN
DoIP
OSI layer 4 Transport
ISO 15765-2 DoCAN transport protocol and network layer services
ISO 15765-4 DoCAN requirements for emissions-related systems
ISO 27145-4 WWH-OBD connection between vehicle and test equipment
ISO 13400-2 DoIP transport protocol and network layer services
OSI layer 3 Network
OSI layer 2 Data link
ISO 11898-1 CAN data link layer and physical signalling
ISO 11898-2 CAN high-speed medium access unit
ISO 13400-3 DoIP wired vehicle interface based on IEEE 802.3
OSI layer 1 Physical

Figure 2 — Reference documents for implementation of WWH-OBDonCAN and WWH-OBDonIP according to the OSI model

6 Unified diagnostic services (UDS) applicable to WWH-OBD

6.1 General

This clause defines how the diagnostic services defined in ISO 14229-1 apply to ISO 27145. For each applicable service, the applicable sub-function and data parameters are defined.

Subclauses 6.3 to 6.8 define additional requirements and/or restrictions for the ISO 14229-1 services that are supported for WWH-OBD in this part of ISO 27145.

NOTE The sub-function parameter definitions take into account that the most significant bit is used for the suppressPosRspMsgIndicationBit parameter, as defined in ISO 14229-1.

6.2 UDS on WWH-OBD overview

This part of ISO 27145 applies the diagnostic services defined in ISO 14229-1 for WWH-OBD-compliant implementations. Table 2 references the ISO 14229-1 services that apply to UDS implementations which meet WWH-OBD requirements.

Table 2 contains all services that apply from ISO 14229-1. For each service, the required support for sub-functions and data parameters is defined. Implementation of additional services from ISO 14229-1, that are not listed in Table 2, is entirely at the discretion of the implementer, except where local regulations define additional requirements. Additional detail for each service is given by the subclause referenced in the rightmost column of Table 2.

Table 2 — Overview of applicable ISO 14229-1 UDS and data ranges

UDS name (ISO 14229-1)	SID value	SFID value	Sub-function name	Comment	Ref.
Data transmission functional unit					
ReadDataByIdentifier	0x22	—	N/A	This service provides read capabilities for static and dynamic data.	6.5
Stored data transmission functional unit					
ReadDTCInformation	0x19	0x04	reportDTCSnapshotRecordBy-DTCNumber	This service provides read capabilities for DTC information.	6.6
		0x06	reportDTCExtendedData-RecordByDTCNumber	The sub-functions are mandatory for WWH-OBD-compliant servers.	
		0x42	reportWWHOBDDTCByMask-Record		
ClearDiagnostic-Information	0x14	—	N/A	This service provides clear DTC information capability. To clear emissions system group information, the parameter groupOfDTC = 0xFFFF33.	6.7
Remote activation of routine functional unit					
RoutineControl	0x31	0x01	startRoutine	This service provides control capability for routines. This sub-function is mandatory for WWH-OBD-compliant servers.	6.8

6.3 Electronic control unit (ECU) response message length too long

If the ECU response message exceeds the length supported by the underlying network layer that has been implemented, a negative response code 0x14 shall be sent by the ECU instead of a positive response message.

6.4 Message byte order

Alphanumeric data derived from SAE J1979-DA and SAE J2012-DA shall be transmitted with the most significant byte first (MSB).

Unless otherwise specified, alphanumeric characters shall conform to the ISO Latin 1 ASCII character set as specified in ISO 27145-2.

See ISO 27145-2 for the message byte order for DTCs according to SAE J1939 Companion Spreadsheet (SPN) and SAE J1939-73:2010, Appendix A (FMI).

6.5 ReadDataByIdentifier (0x22) service

6.5.1 General

This part of ISO 27145 complies with the requirements of ISO 14229-1. Subclause 6.5.2 defines additional requirements or imposes restrictions applicable to the service referenced.

6.5.2 WWH-OBD-specific requirements

6.5.2.1 Requirement — Minimum number of DIDs per request to be supported

Table 3 specifies a minimum number of DIDs per request that shall be supported by the server(s) as part of a request message of the service ReadDataByIdentifier, even if the server(s) might not have support for the data referenced by the DID(s).

Table 3 — Requirement definition: Minimum number of DIDs per request to be supported by a server

Requirement name	**Minimum number of DIDs per request to be supported by a server**
Affects	WWH-OBD server(s)
Brief description	The ReadDataByIdentifier service allows the client to request data record values from the server(s) identified by one or more DIDs. The client request message contains one or more DataIdentifier values that identify data record(s) maintained by the server (see ISO 27145-2). Upon receiving a ReadDataByIdentifier request, the server shall access the data elements of the records specified by the DID parameter(s) and transmit their value in one single ReadDataByIdentifier positive response message containing the associated dataRecord parameter(s).
Requirement	The server shall support at least six DIDs simultaneously in a request and response message if requested by the external test equipment.

6.6 ReadDTCInformation (0x19) service

6.6.1 General

This part of ISO 27145 complies with the requirements of ISO 14229-1. Subclause 6.6.2 defines additional requirements or imposes restrictions applicable to the service referenced.

6.6.2 WWH-OBD-specific requirements

6.6.2.1 Requirement — DTC format identification

Table 4 specifies the two DTC formats which shall be supported by the GTR WWH-OBD-compliant server(s) and external test equipment.

Table 4 — Requirement definition: DTC format identification

Requirement name	**DTC format identification**
Affects	Client(s), WWH-OBD server(s)
Brief description	The DTCFormatIdentifier defined in ISO 14229-1 is a 1-byte parameter value which defines the format of a DTC reported by the server.
Requirement	DTCs reported by services and sub-functions in accordance with ISO 27145 shall always use only one of the two formats specified in ISO 27145-2: — SAE_J2012-DA_DTCFormat_04: This parameter value identifies the WWH-OBD DTC format reported by the server, as defined in the SAE J2012-DA specification. — SAE_J1939-73_DTCFormat: This parameter value identifies the DTC format reported by the server, as defined in the SAE J1939-73 specification. The values of the DTCFormatIdentifier are specified in ISO 14229-1.

6.6.2.2 Requirement — Support of DTCStatusAvailabilityMask parameter

Table 5 specifies the bits which are defined in the same way as for statusOfDTC and which represent the status bits that are supported by GTR WWH-OBD-compliant server(s). Bits that are not supported by the server(s) shall be set to 0.

Table 5 — Requirement definition: Support of DTCStatusAvailabilityMask parameter

Requirement name	**Support of DTCStatusAvailabilityMask parameter**
Affects	WWH-OBD server(s)
Brief description	The DTCStatusAvailabilityMask parameter indicates the statusOfDTC bits supported by GTR WWH-OBD-compliant server(s). NOTE It is up to the manufacturer to support additional bits per DTC as deemed necessary to fully support their diagnostics.

Requirement	The GTR WWH-OBD-compliant server shall support at least the following statusOfDTC parameter bits per DTC:
	— bit 2, pendingDTC (PDTC);
	— bit 3, confirmedDTC (CDTC);
	— bit 4, testNotCompletedSinceLastClear (TNCSLC);
	— bit 6, testNotCompletedThisOperationCycle (TNCTOC).
	The values of the DTCStatusAvailabilityMask parameter are specified in ISO 14229-1.

6.6.2.3 Requirement — Support of DTCSeverityAvailabilityMask parameter

Table 6 specifies the bits which are defined in the same way as for DTCSeverity and DTC Class and which represent the status bits that are supported by GTR WWH-OBD-compliant server(s). Bits that are not supported by the server(s) shall be set to 0.

Table 6 — Requirement definition: Support of DTCSeverityAvailabilityMask parameter

Requirement name	**Support of DTCSeverityAvailabilityMask parameter**
Affects	WWH-OBD server(s)
Brief description	The DTCSeverityAvailabilityMask parameter indicates the DTC Class bits supported by GTR WWH-OBD-compliant server(s).

Requirement	The GTR WWH-OBD-compliant server shall support at least the following DTC Class parameter bits:
	— bit 1, Class_1 (Class A);
	— bit 2, Class_2 (Class B1);
	— bit 3, Class_3 (Class B2);
	— bit 4, Class_4 (Class C).
	The values of the DTC Class parameter are specified in ISO 14229-1.

6.6.2.4 Requirement — Request mask handling of DTC Severity Mask Record

The requirements specified in this section allow for the implementation of the malfunction classification concept defined in the GTR WWH-OBD, Module B, section 4.5 (see Reference [17]).

Table 7 specifies the WWH-OBD GTR DTC masking requirements. The following definitions apply:

— AND, OR	logical operation;
— &	bitwise AND;
— &&	logical AND;
— ==	equal to (comparison operator);
— =	assignment operator;
— !=	not equal to.

6.6.2.5 GTR WWH-OBD DTC status diagram

Figure 3 shows an implementation example of the GTR WWH-OBD DTC state timings depending on operation cycle and monitoring results. Additionally, the diagram describes related states of the DTC status bits defined by ISO 14229-1/ISO 27145. Finally, the diagram illustrates how the client displays GTR WWH-OBD-compliant DTC states based on the ISO 14229-1 status bit definitions. Reusing the DTC status bit definitions forces a specific implementation of the GTR WWH-OBD DTC status requirements but fully satisfies them.

The following conditions apply:

- A DTC will be reported as “Confirmed & Active” for this operation sequence where the malfunction occurs again after the DTC has been reported as “Previously Active”.
- Since the DTC status is changed from “Previously Active” to “Confirmed & Active” one operation sequence earlier than required, presuming the DTC was in “Previously Active” state, the related MI (short/continuous) illuminates one operation sequence prior to when required.
- The warm-up cycle counter indicating the number of cycles during which a DTC was in “Previously Active” status (aging counter) counts upward from zero (0) only if the related MI has been de-activated.

The implementation shown in Figure 3 is an example and for illustration purposes only. It shows the discriminatory malfunction indicator (MI) for Class B2 malfunctions.

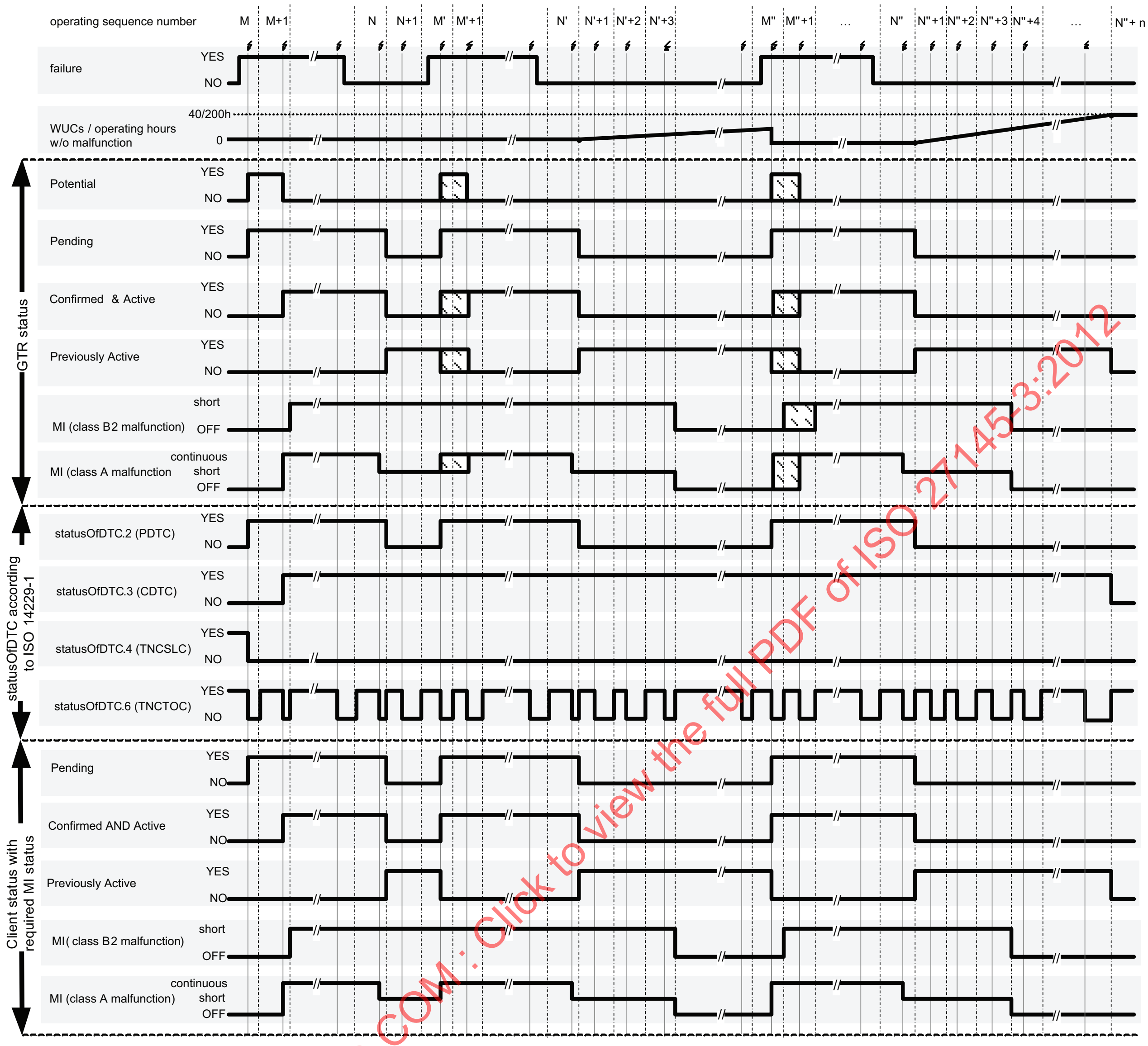
When mapping the GTR DTC status to DTC status bits (statusOfDTC) defined according to ISO 14229-1, the definitions in Table 8 apply.

Table 7 — Requirement definition: Request mask handling of DTC Severity Mask Record

Requirement name	**Request mask handling of DTC Severity Mask Record**
Affects	Client(s), server(s)
Brief description	The DTC Severity Mask Record consists of three different request mask elements, namely the DTC Severity Mask byte and the DTC Status Mask, where the DTC Severity Mask byte is separated into DTC Severity Mask (bits 5-7) and DTC Class Mask (bits 0-4). Each of these three mask elements needs to be processed separately for each individual DTC that an ECU supports. A DTC is only included in the positive response to a sub-function 0x42 request if at least the processing of the DTC Status Mask and the DTC Class Mask yielded a non-zero result.
Pre-requisites	— Each individual WWH-OBD-relevant DTC is assigned to a unique fault class (A, B1, B2, C). This means that, for each DTC of functional group 0x33, an ECU shall set one bit out of the four dedicated DTC class bits to one (1), indicating that the respective DTC belongs to one unique fault class. — When requesting DTCs by sub-function 0x42, the DTC Severity Mask (bit 5-7) shall only be considered if at least one bit of this mask element is set to one.
Requirement	If the server does not implement DTC Severity, the GTR WWH-OBD-compliant server/ECU shall perform a bitwise logical ANDing of — the DTC Status Mask and the actual DTC Status, and — the requested DTC Class Mask and the actual DTC class for each DTC upon request of the test equipment. The positive response message shall include only those DTCs for which both ANDing operations yielded a non-zero result. **(((DTCStatusMask & statusOfDTC) !=0) && ((DTCClassMask & actual DTC class) !=0))** == **TRUE** If the server implements DTC Severity, the GTR WWH-OBD-compliant server/ECU shall perform a bitwise logical ANDing of — the DTC Status Mask and the actual DTC Status, — the requested DTC Severity Mask and the actual DTC Severity, and — the requested DTC Class Mask and the actual DTC class for each DTC upon request of the test equipment. The positive response message shall include only those DTCs for which all three ANDing operations yielded a non-zero result. **(((DTCStatusMask & statusOfDTC) !=0) && ((DTCClassMask & actual DTC class) !=0) && ((DTCSeverityMask & actual DTC Severity) !=0))** == **TRUE**

Table 8 — Mapping of GTR WWH-OBD DTC status and ISO 14229-1 statusOfDTC

DTC status according to GTR WWH-OBD	**DTC status bits (statusOfDTC) according to ISO 14229-1**	
Potential	Pending AND (NOT Confirmed)	statusOfDTC.2 (PDTC) AND [NOT statusOfDTC.3 (CDTC)]
Pending	Pending	statusOfDTC.2 (PDTC)
Confirmed AND Active	Confirmed AND Pending	statusOfDTC.3 (CDTC) AND statusOfDTC.2 (PDTC)
Previously Active	Confirmed AND (NOT Pending)	statusOfDTC.3 (CDTC) AND [NOT statusOfDTC.2 (PDTC)]



Key

ϟ point at which monitoring of the concerned malfunction occurs

▨ status defined by the manufacturer as allowed by the GTR

// operating sequence

M first operating sequence during which a malfunction is detected

N first operating sequence during which no malfunction is detected

M′, N′ second operating sequence during which a malfunction is detected/no malfunction is detected

M″, N″ third operating sequence during which a malfunction is detected/no malfunction is detected

Figure 3 — GTR WWH-OBD DTC status diagram

Table 9 defines examples based on the minimum requirements for how the client retrieves one or multiple GTR WWH-OBD DTC(s) of a specific status and class. This table covers all use cases required by the GTR WWH-OBD, Module B. The request message has been defined under the assumption that the server does not support DTC Severity (i.e. bits 7–5 = 0).

Table 9 — WWH-OBD DTC retrieval examples

DTC status and class as defined in the use cases of ISO 27145-1	Request (Req.) and Response (Resp.) message		DTCStatus-Mask	DTC Severity-Mask byte
			bits 7 ... 0	bits 7 ... 0
Confirmed and Active DTCs for Class A malfunctions DTC format = SAE_J2012-DA_DTCFormat_04 The client shall identify the DTCs matching the confirmed and active status – remove all other DTCs from the result list. See Figure 4 for a description of client behaviour.	Req.	0x19, 0x42, 0x33, 0x08, 0x02	0000 1000$_b$	0000 0010$_b$
	Resp.	0x59, 0x42, 0x33, 0x5C, 0x1E, 0x04, [0x02, DTCHB, DTCMB, DTCLB, SODTC] ..., […]		
Confirmed and Active DTCs for Class B (B1 and B2) malfunctions DTC format = SAE_J2012-DA_DTCFormat_04 The client shall identify the DTCs matching the confirmed and active status – remove all other DTCs from the result list. See Figure 5 for a description of client behaviour.	Req.	0x19, 0x42, 0x33, 0x08, 0x0C	0000 1000$_b$	0000 1100$_b$
	Resp.	0x59, 0x42, 0x33, 0x5C, 0x1E, 0x04, [0x0C, DTCHB, DTCMB, DTCLB, SODTC] ..., […]		
Confirmed and Active DTCs for Class C malfunctions DTC format = SAE_J2012-DA_DTCFormat_04 The client shall identify the DTCs matching the confirmed and active status – remove all other DTCs from the result list. See Figure 6 for a description of client behaviour.	Req.	0x19, 0x42, 0x33, 0x08, 0x10	0000 1000$_b$	0001 0000$_b$
	Resp.	0x59, 0x42, 0x33, 0x5C, 0x1E, 0x04, [0x10, DTCHB, DTCMB, DTCLB, SODTC] ..., […]		
Pending DTCs and their associated class DTC format = SAE_J2012-DA_DTCFormat_04 The client shall identify the DTCs matching the pending status – remove all other DTCs from the result list. See Figure 7 for a description of client behaviour.	Req.	0x19, 0x42, 0x33, 0x04, 0x1E	0000 0100$_b$	0001 1110$_b$
	Resp.	0x59, 0x42, 0x33, 0x5C, 0x1E, 0x04, [DTCS, DTCHB, DTCMB, DTCLB, SODTC] ..., […]		
Previously Active DTCs and their associated class DTC format = SAE_J2012-DA_DTCFormat_04 The client shall identify the DTCs matching the previously active status – remove all other DTCs from the result list. See Figure 8 for a description of client behaviour.	Req.	0x19, 0x42, 0x33, 0x08, 0x1E	0000 1000$_b$	0001 1110$_b$
	Resp.	0x59, 0x42, 0x33, 0x5C, 0x1E, 0x04, [DTCS, DTCHB, DTCMB, DTCLB, SODTC] ..., […]		

Figures 4 to 8 illustrate the definition of the client's request message and the server's action in order to provide the requested DTC information in a positive response message.

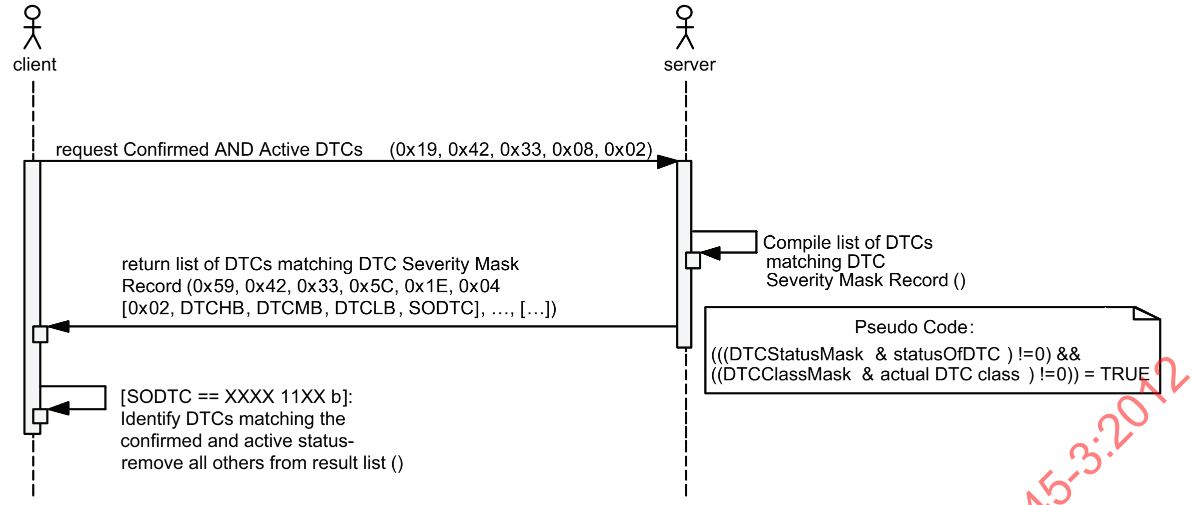

client
server
request Confirmed AND Active DTCs (0x19, 0x42, 0x33, 0x08, 0x02)
Compile list of DTCs matching DTC Severity Mask Record ()
return list of DTCs matching DTC Severity Mask Record (0x59, 0x42, 0x33, 0x5C, 0x1E, 0x04 [0x02, DTCHB, DTCMB, DTCLB, SODTC], …, […])
Pseudo Code:
(((DTCStatusMask & statusOfDTC) !=0) &&
((DTCClassMask & actual DTC class) !=0)) = TRUE
[SODTC == XXXX 11XX b]: Identify DTCs matching the confirmed and active status- remove all others from result list ()

Figure 4 — Retrieval of Confirmed AND Active DTCs of Class A malfunctions

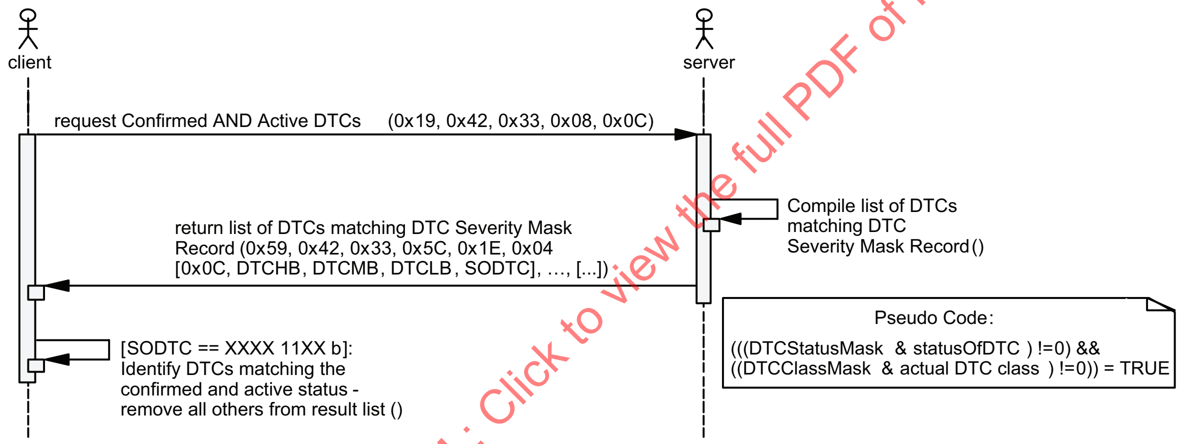

client
server
request Confirmed AND Active DTCs (0x19, 0x42, 0x33, 0x08, 0x0C)
Compile list of DTCs matching DTC Severity Mask Record()
return list of DTCs matching DTC Severity Mask Record (0x59, 0x42, 0x33, 0x5C, 0x1E, 0x04 [0x0C, DTCHB, DTCMB, DTCLB, SODTC], …, [...])
Pseudo Code:
(((DTCStatusMask & statusOfDTC) !=0) &&
((DTCClassMask & actual DTC class) !=0)) = TRUE
[SODTC == XXXX 11XX b]: Identify DTCs matching the confirmed and active status - remove all others from result list ()

Figure 5 — Retrieval of Confirmed AND Active DTCs of Class B1 and B2 malfunctions

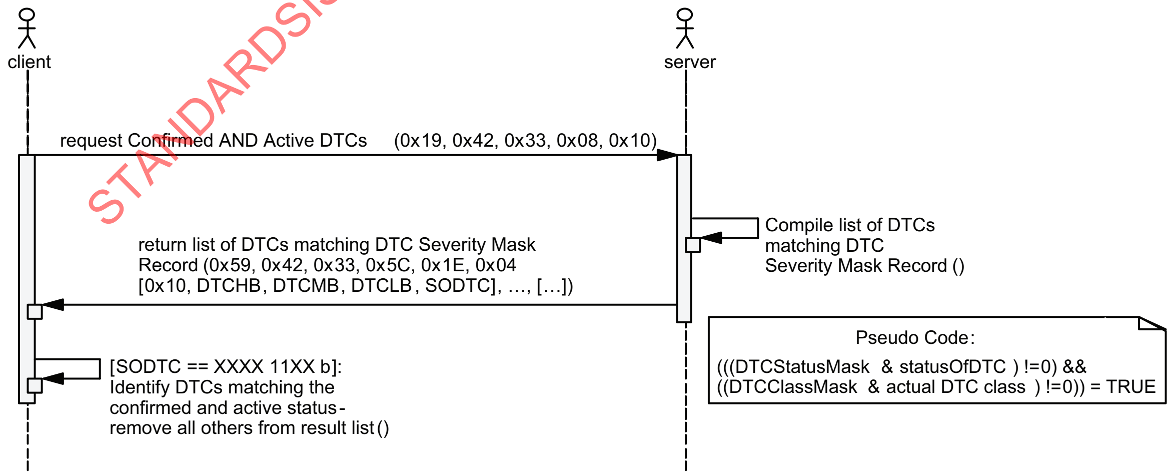

client
server
request Confirmed AND Active DTCs (0x19, 0x42, 0x33, 0x08, 0x10)
Compile list of DTCs matching DTC Severity Mask Record ()
return list of DTCs matching DTC Severity Mask Record (0x59, 0x42, 0x33, 0x5C, 0x1E, 0x04 [0x10, DTCHB, DTCMB, DTCLB, SODTC], …, […])
Pseudo Code:
(((DTCStatusMask & statusOfDTC) !=0) &&
((DTCClassMask & actual DTC class) !=0)) = TRUE
[SODTC == XXXX 11XX b]: Identify DTCs matching the confirmed and active status- remove all others from result list()

Figure 6 — Retrieval of Confirmed AND Active DTCs of Class C malfunctions

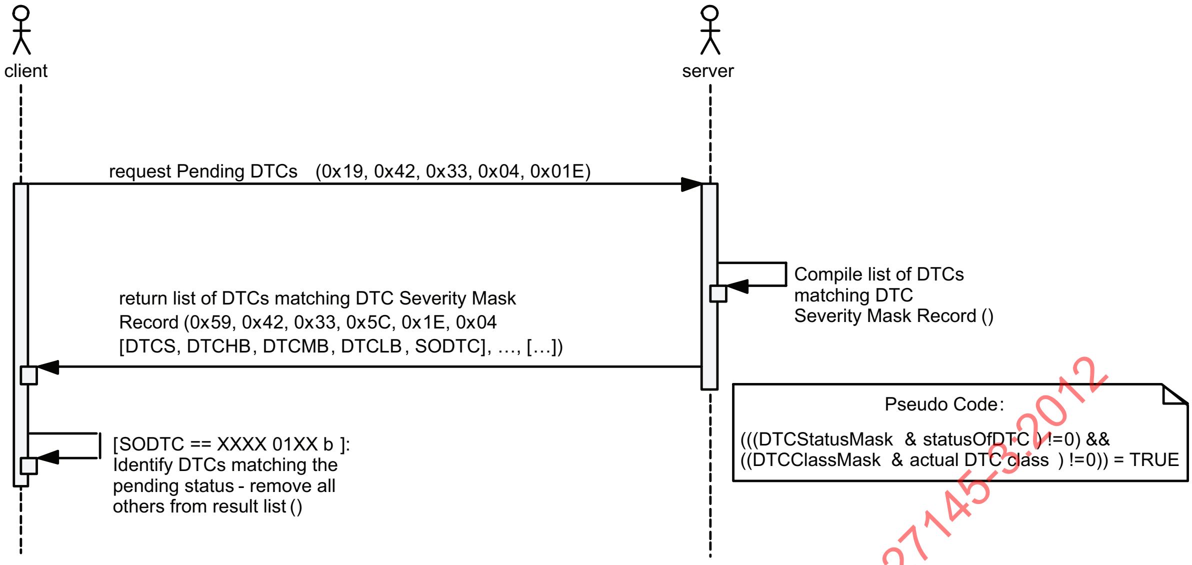

client
server
request Pending DTCs (0x19, 0x42, 0x33, 0x04, 0x01E)
Compile list of DTCs matching DTC Severity Mask Record ()
return list of DTCs matching DTC Severity Mask Record (0x59, 0x42, 0x33, 0x5C, 0x1E, 0x04 [DTCS, DTCHB, DTCMB, DTCLB, SODTC], ..., [...])
Pseudo Code:
(((DTCStatusMask & statusOfDTC) !=0) &&
((DTCClassMask & actual DTC class) !=0)) = TRUE
[SODTC == XXXX 01XX b]: Identify DTCs matching the pending status - remove all others from result list ()

Figure 7 — Retrieval of Pending DTCs and their associated class

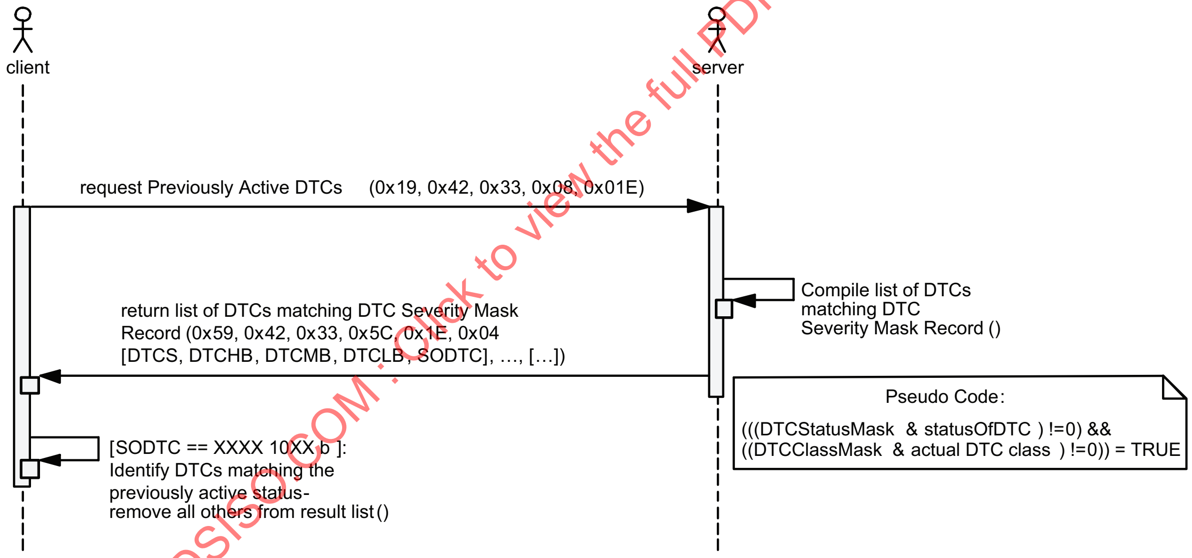

client
server
request Previously Active DTCs (0x19, 0x42, 0x33, 0x08, 0x01E)
Compile list of DTCs matching DTC Severity Mask Record ()
return list of DTCs matching DTC Severity Mask Record (0x59, 0x42, 0x33, 0x5C, 0x1E, 0x04 [DTCS, DTCHB, DTCMB, DTCLB, SODTC], ..., [...])
Pseudo Code:
(((DTCStatusMask & statusOfDTC) !=0) &&
((DTCClassMask & actual DTC class) !=0)) = TRUE
[SODTC == XXXX 10XX b]: Identify DTCs matching the previously active status- remove all others from result list ()

Figure 8 — Retrieval of Previously Active DTCs and their associated class

6.6.2.6 Requirement — FunctionalGroupIdentifier

Table 10 specifies the purpose and requirements of the FunctionalGroupIdentifier.

Table 10 — Requirement definition: FunctionalGroupIdentifier

Requirement name	**FunctionalGroupIdentifier**
Affects	Client(s), server(s)
Brief description	The FunctionalGroupIdentifier has been introduced to distinguish commands sent by the test equipment between different functional system groups within an electrical architecture which consists of many different servers. If a server has implemented software of the emissions system as well as other systems which may be inspected during an I/M test, it is important that only the DTC information of the requested functional system group is reported. An emissions I/M test should not be failed because another functional system group, e.g. safety system group, has DTC information stored. The values of the FunctionalGroupIdentifier are specified in ISO 14229-1.
Requirement	The FunctionalGroupIdentifier value for the WWH-OBD emissions system group is defined in ISO 14229-1.

6.6.3 WWH-OBD-specific ReadDTCInformation service examples

WWH-OBD-specific ReadDTCInformation service examples are described in A.3.

6.7 ClearDiagnosticInformation (0x14) service

6.7.1 General

This part of ISO 27145 complies with the requirements of ISO 14229-1. Subclause 6.7.2 defines additional requirements or imposes restrictions applicable to the service referenced.

6.7.2 WWH-OBD-specific requirements

6.7.2.1 Requirement — Clear all WWH-OBD DTC information

Table 11 defines which DTC-related information shall be cleared in the server’s memory when receiving the appropriate request message.

Table 11 — Requirement definition: Clear all WWH-OBD DTC information

Requirement name	**Clear all WWH-OBD DTC information**
Affects	All WWH-OBD system servers/ECUs
Brief description	The purpose of the ClearDiagnosticInformation service is to enable test equipment to erase/reset DTC-related information stored/captured by the server after a malfunction of the GTR WWH-OBD-compliant system was detected. This service is used by technicians after performing a system repair. NOTE Depending on the electronic system design, it is not guaranteed that the resetting operations which are linked to the ClearDiagnosticInformation service request will be completed before the vehicle powers down completely, thus a battery supply voltage disconnect immediately after sending the request might result in the information still being available when the vehicle is powered up again.
Requirement	The server shall clear the following information after successful reception of the ClearDTCInformation request message with the groupOfDTC parameter set to “ClearAllWWHOBDDTC” and the FunctionalGroupIdentifier set to “emissions system group”: — malfunction indicator status (value to be reset); — readiness of the OBD system (value to be reset); — number of engine operating hours since activation of the malfunction indicator (continuous MI counter) (data to be erased); — all DTCs (data to be erased); — B1 counter(s) (reset to the value specified in the regulation); — number of engine operating hours from the B1 counter(s) (value to be reset); — freeze frame data (snapshot record data) requested by this module/ECU (data to be erased).

NOTE 1 Other manufacturer-specific “clearing/resetting” actions can also occur in response to this request message.

NOTE 2 The list given in Table 11 is current at the time of publication of this part of ISO 27145. It is the vehicle manufacturer’s responsibility to take into account changes to regulations and additional local requirements in order to address any conflicts between the list above and applicable regulations.

6.7.3 WWH-OBD-specific ClearDiagnosticInformation service examples

WWH-OBD-specific ClearDiagnosticInformation service examples are described in A.4.

6.8 RoutineControl (0x31) service

6.8.1 WWH-OBD-specific requirements

6.8.1.1 General

This part of ISO 27145 complies with the requirements as defined in ISO 14229-1. This subclause defines additional requirements or restrictions applicable to the service referenced.

6.8.1.2 Requirement — Activate on-board control routines and report test results

Table 12 defines how to activate on-board control routines and report test results.

Table 12 — Requirement defintion: Activate on-board control routine and report test results

Requirement name	**Activate on-board control routine and report test results**
Affects	WWH-OBD-compliant ECUs which support control rountines specified in SAE J1979-DA
Brief description	The purpose of the RoutineControl service is to enable external test equipment to start a control routine within the server in order to activate an EVAP test, for example. This service is used by technicians, for example to perform a validation of the EVAP system regarding whether a previously performed repair has been completed with success.
Requirement	The server shall start the execution of the requested control routine if the vehicle meets all conditions required to perform the control routine (e.g. engine running/engine not running). The RoutineControl service can only be commanded by the external test equipment if supported by the server (RIDs supported). The sub-function startRoutine shall be supported by RoutineControl service. The RoutineInfo byte is mandatory for any routine where the routineStatusRecord is defined by the SAE J1979-DA (i.e. defined in routineStatusRecord of RID 0xE000 – 0xE1FF) even if the defined size of the routineStatusRecord equals zero (0) data bytes. The definition of the RoutineInfo byte shall be left to the vehicle manufacturer and shall not be interpreted by the WWH-OBD-compliant external test equipment. For a detailed definition of the RoutineControl service, see ISO 14229-1.

6.8.2 RoutineControl (0x31) service examples

WWH-OBD-specific RoutineControl service examples are described in A.2.

7 Application layer requirements

7.1 Application layer services

This part of ISO 27145 uses the application layer services defined in ISO 14229-1 for client-server based systems, to perform functions such as test, inspection, monitoring or diagnosis of on-board vehicle servers.

7.2 Application layer protocol

This part of ISO 27145 uses the application layer protocol defined in ISO 14229-1.

7.3 Addressing and timing requirements

7.3.1 General

To ensure a clear understanding of the data-link-dependent application layer timings, this document differentiates between two major timing concepts.

For WWH-OBD-compliant systems using the data link defined in ISO 15765 (DoCAN), the response time is measured at the beginning of the reception of a response message (see the definition of timing parameter P2 given in Table 15).

For WWH-OBD-compliant systems using the data link defined in ISO 13400 (DoIP), the response time is measured when the response has been completely received (see the definition of timing parameter P6 given in Table 14).

Consequently, the values to be defined for P6 depend on the response length of the message to be received.

Although the P2 values compared to the P6 values seem to imply a faster application layer timing because the values are smaller, the total time needed to transfer a complete response to the test equipment will generally be faster on DoIP, although the maximum P6 value is initially set to a longer value than the P2 value.

ISO 13400 DoIP $P6_{Client}$ timeout differs (longer timeout) from that given in ISO 15765-4 DoCAN. (See the calculations defined in Table 14 and Table 15.)

NOTE All message timing definitions assume that there is no additional processing time needed for passing data and status information between the individual OSI layers. Thus if the test equipment is running on an operating system that introduces processing delays between the individual OSI layer software stacks, this needs to be taken into account separately by the external test equipment application; it is not specified in this part of ISO 27145.

7.3.2 GTR WWH-OBD use cases and addressing methods

Table 13 specifies the GTR WWH-OBD use cases and addressing methods of a WWH-OBD-compliant server/ECU in order to achieve consistent timing behaviour between WWH-OBD-compliant servers/ECUs and WWH-OBD-compliant external test equipment.

Table 13 — GTR WWH-OBD use cases and addressing methods

Requirement name	**GTR WWH-OBD use cases and addressing methods**		
Affects	Client/external test equipment based on ISO 27145-6		
Brief description	The purpose of this requirement is to define message length constraints for individual WWH-OBD use cases in order to achieve consistent response timing behaviour between WWH-OBD-compliant servers/ECUs and WWH-OBD-compliant external test equipment. The transmission time of a response message from a server/ECU depends on the payload of the message [length of protocol data unit (PDU)]. The more data bytes included in a response message, the more transmission time is needed.		
Applicability	**Applicable use cases**	**Functional addressing**	**Physical addressing**
	a) Protocol-supported identification request [SID: 0x22, DID: 0xF810: Read out “protocol identification” (0x01 = ISO 27145-4)]	supported by server/ECU	optional
	b) VIN (vehicle identification number) [SID: 0x22, DID: 0xF802]	supported by server/ECU	optional
	c) ISO 27145-1 use case #1 **Vehicle roadworthiness**. The WWH-OBD vehicle OBD system information includes consolidated and packeted data items about the vehicle’s roadworthiness status to support a functionally addressed request and a single response message including all required data items from one server/ECU at the roadside [SID: 0x22; DID: 0xF490] which contains — discriminatory/non-discriminatory display strategy, — presence of a continuous MI, — readiness status of the OBD system, and — number of engine operating hours for which the continuous MI was last activated (continuous MI counter).	supported by server/ECU	optional
	d) Clear all WWH-OBD DTCs in all WWH-OBD servers/ECUs of a specific vehicle [0x14 0xFF 0xFF 0x33]	supported by server/ECU	optional
	e) WWH-OBD ECU OBD system information includes packeted data items about the WWH-OBD-compliant ECU’s roadworthiness status in order to support a physically addressed request and a single response message including all required data items from that server/ECU [SID: 0x22; DID: 0xF491] which contains — discriminatory/non-discriminatory display strategy, — presence of a continuous MI, — number of engine operating hours for which the continuous MI was last activated (continuous MI counter), and — highest ECU B1 counter.	optional	supported by server/ECU
	f) All WWH-OBD data retrieval except the use case in which functional addressing applies.	optional	supported by server/ECU
	IMPORTANT — a) through c) shall only be requested as a single DID per request message, in order to comply with the server/ECU response performance requirement (see Table 14 and Table 15). NOTE All data items are specified in SAE J1979-DA.		
Requirement	— **Functional addressing** shall be used by the external test equipment only if use cases a) through d) apply. — **Physical addressing** shall be used by the external test equipment if use case e) or f) applies.		

7.3.3 ISO 13400 DoIP message timing definition for WWH-OBD use cases

The message timing definition specified in this subclause shall ensure that a WWH-OBD GTR-compliant vehicle can respond within its response performance required.

The WWH-OBD message timing definition for the default diagnostic session shall be in accordance with Table 14.

Table 14 — Message timing definition for ISO 13400 DoIP in defaultSession

Timing parameter	Definition	Minimum [ms]	Maximum [ms]
$\Delta P6$ [a]	The $\Delta P6$ parameter is defined to be the worst-case vehicle-network-design-dependent message transmission delay, such as delays introduced by gateways and bus-load arbitration delay. The value of $\Delta P6$ is divided between the time to transmit the request to the addressed server/ECU ($\Delta P6_{request}$) and the time to transmit the response to the client/tester ($\Delta P6_{response\_part1}$). It also depends on the request/response message length ($\Delta P6_{response\_part2}$).	> 0	50 to 4 950
$P2_{Server}$	The $P2_{Server}$ parameter is a performance requirement for the server/ECU to start with the response message after reception of a request message.	0	50
$P6_{Client}$ [b]	The $P6_{Client}$ parameter timeout is in order for the client to wait after the successful transmission of a request for the start of incoming response messages. $P6_{Client\_min} = P2_{Server\_max} + \Delta P6_{max}$	100 to 5 000 [c]	— [d]
$P2^*_{Server}$	The $P2^*_{Server}$ parameter is a performance requirement for the server to start with the response message after the transmission of a negative response message with the negative response code 0x78 (enhanced response timing).	0 [e]	5 000
$P6^*_{Client}$ [b]	The $P6^*_{Client}$ parameter is the enhanced timeout in order for the client to wait after reception of a negative response message with the negative response code (NRC) 0x78 for the start of incoming response messages ($P2^*_{Server\_max} + \Delta P2_{max}$). $P6^*_{Client\_min} = P2^*_{Server\_max} + \Delta P6_{max}$	5 050 to 9 950 [f]	— [g]
$P3_{Client\_Phys}$	The $P3_{Client\_Phys}$ parameter is the minimum time for the client to wait after successful transmission of a physically addressed request message with no response required before the next physically addressed request message can be transmitted. $P3_{Client\_Phys} = P2_{Server\_max} + \Delta P6_{max}$	100	— [h]
$P3_{Client\_Func}$	The $P3_{Client\_Func}$ parameter is the minimum time for the client to wait after successful transmission of a functionally addressed request message before the next functionally addressed request message can be transmitted, in the case that no response is required or that the requested data is only supported by a subset of the functionally addressed servers. $P3_{Client\_Func} = P2_{Server\_max} + \Delta P6_{max}$ Note that even if no response is required, the server can send an NRC 0x78 response in the case that the execution of the requested service takes more time and is not possible within $P2_{Server}$ (50 ms), followed by the final response. If this scenario applies, the client shall enable $P6^*_{client}$ timeout handling.	100	— [h]

Table 14 (*continued*)

[a] The parameter $\Delta P6$ takes into account any (worst-case) vehicle-network-design-dependent message transmission delays and depends on the request/response length: $\Delta P6 = \Delta P6_{Request} + \Delta P6_{Response}$

These delays, introduced for example by gateways, baudrates, transmission times and safety margins, can be mapped on a length-dependent dynamic value:

$\Delta P6 = \Delta P6_{Request} + \Delta P6_{Response}$

$\Delta P2 = \Delta P6_{Request} + \Delta P6_{Response\_part1} + \Delta P6_{Response\_part2} = 50$ ms + [0,85 ms/byte × (response length − 6 bytes)]

where

$\Delta P6_{Request} = 30$ ms fix value for short requests;

$\Delta P6_{Response\_part1} = 20$ ms fix value for short responses;

$\Delta P6_{Response\_part2} =$ [0,85 ms/byte ×(response length − 6 bytes)];

$\Delta P6_{Response} = 20$ ms + [0,85 ms/byte ×(response length − 6 bytes)];

with response length ≥ 6 bytes.

The factor 0,85 ms/byte is based on typical values for system design parameters like separation time and baudrate.

The data length factor (response length − 6 bytes) represents the gain of transmission time due to rising response length (PDU data).

— Minimum value $\Delta P6_{min} \approx 50$ ms (short request with short response and response length ≤ 6 bytes).

— Maximum value $\Delta P6_{max} \approx$ 4 950°ms (4 095 bytes PDU).

Based on this calculation the following performance requirement depending on the request/response message length shall be fulfilled by the vehicle:

$\Delta P6 \leq 50$ ms for all requests with response data length (PDU) ≤ 6 bytes;

$\Delta P6 \leq 50$ ms + [0,85 ms/byte × (response length − 6 bytes)] for all requests with response data length (PDU) > 6 bytes.

[b] In ISO 13400 DoIP, the timer for $P6_{Client}$ is started by receiving the Confirmation Acknowledge of the DoIP gateway. $P6_{Client}$ and $P6^*_{Client}$ are stopped when the complete message is received.

[c] The $P6_{Client}$ timeout value depends on the addressing method and the kind of data requested:

— generic timeout for physical request: $P6_{Client} = 5\,000$ ms;

— timeout for functional request for identification with unknown number of responses: $P6_{Client} = 100$ ms;

— timeout for selected functional requests with known number of short requests and responses: $P6_{Client} = 100$ ms.

Negative responses, including the first response with response code 0x78, follow the performance requirement for the ECU and are expected to arrive at the external test equipment within $P6_{Client} = 100$ ms.

[d] The maximum value used for $P6_{Client}$ to wait for complete reception of the corresponding response message is left to the discretion of the client as long as it is greater than the specified minimum value of $P6_{Client}$. However, the vehicle has to respond within the minimum value of $P6_{Client}$.

[e] During enhanced response timing, the minimum time between transmission of consecutive negative response messages (each with negative response code 0x78) shall be 0,3 × $P2^*_{Server\_max}$ in order to avoid flooding the data link with unnecessary negative response code 0x78 messages.

[f] The minimum value of $P6^*_{Client}$ shall be chosen according to the definition for $P6_{Client}$:

— functional requests: 5 050 ms;

— physical requests: 9 950 ms.

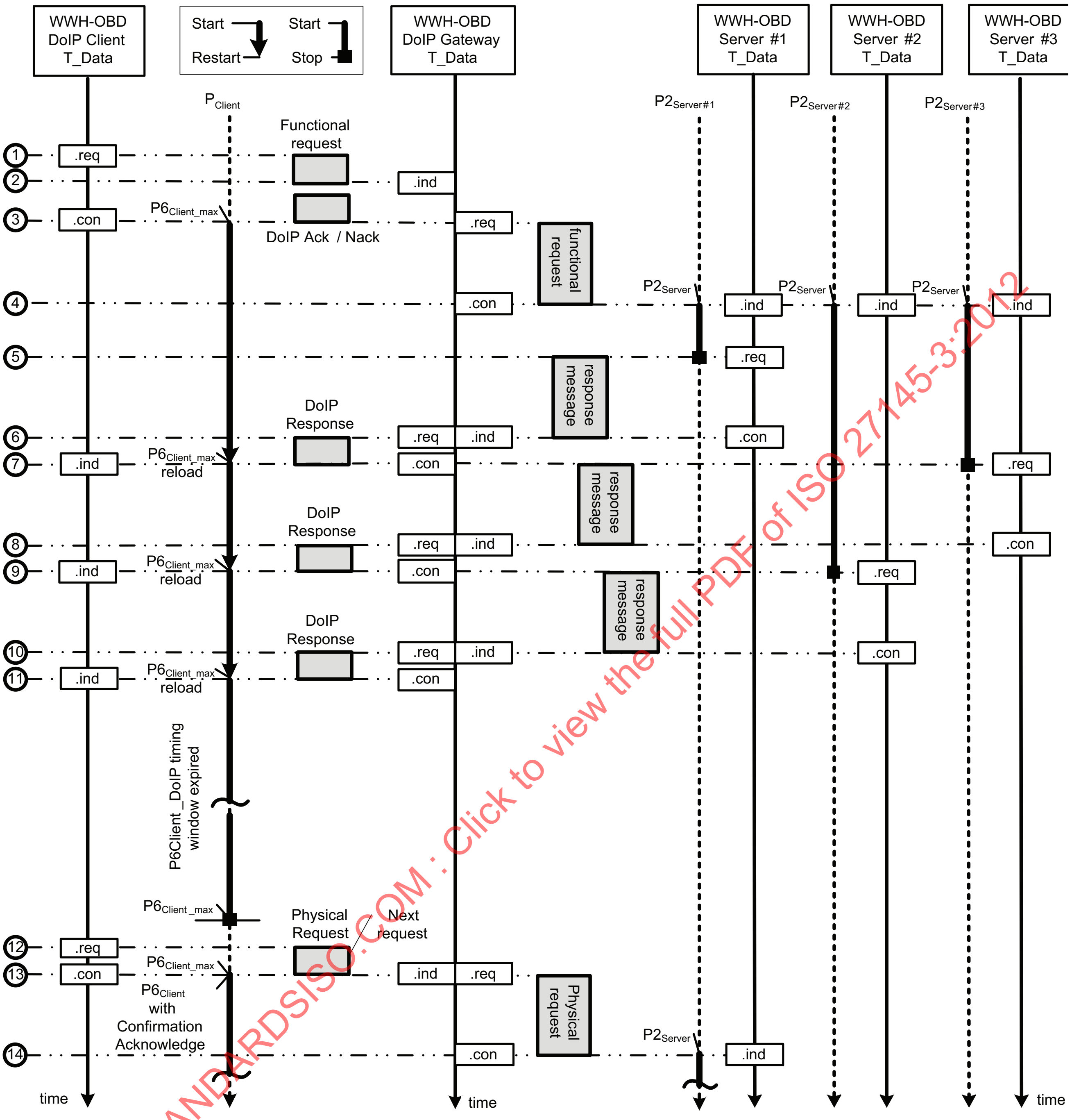
[g] The maximum value a client uses for $P6^*_{Client}$ to wait for complete reception of the corresponding response message is left to the discretion of the client as long as it is greater than the specified minimum value of $P6^*_{Client}$. However, the vehicle has to respond within the minimum value of $P6^*_{Client}$.

[h] The maximum time a client waits until it transmits the next request message is at the discretion of the client.

IMPORTANT — For WWH-OBD, the timing definitions in this part of ISO 27145 take precedence over other related documents.

Figure 9 illustrates a functional request message from the client to the DoIP gateway and which the DoIP gateway forwards on to the WWH-OBD-compliant CAN network. The WWH-OBD-compliant servers send response messages which the DoIP gateway transmits to the client. The same sequence also applies to the physical request from the client to the vehicle.

NOTE The CAN network as part of the entire vehicle network in Figure 9 is an example only.



Key

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

2 **Gateway DoIP T_Data.ind**: network layer issues to gateway application the indication of the reception of a request message.

3 **Client T_Data.con**: network layer issues to diagnostic application the confirmation of the completion of the functionally addressed request message. Client starts the P_{Client} timer with $P_{Client} = P6_{Client\_max}$.
Gateway DoIP: application issues an Acknowledge/No Acknowledge frame to the DoIP client.
Gateway DoCAN T_Data.req: gateway application forwards the functionally addressed request by issuing a T_Data.req to the DoCAN network layer.

4 **Gateway DoCAN T_Data.con**: network layer issues to gateway application the confirmation of the completion of the request message.
Servers #1, #2, and #3 DoCAN T_Data.ind: network layer issues to diagnostic application the indication of the reception of a request message. Servers #1, #2, and #3 start their $P2_{Server}$ timers using the default reload value $P2_{Server} = P2_{Server\_max}$.

5 **Server #1 DoCAN T_Data.req**: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{Server}$. Server #1 stops the $P2_{Server}$ timer.

6 **Server #1 DoCAN T_Data.con**: network layer issues to diagnostic application the confirmation of the completion of the response message.
Gateway DoCAN T_Data.ind: network layer issues to gateway application the indication of the reception of the response message.
Gateway DoIP_Data.req: gateway application forwards the response by issuing a T_data.req to the DoIP network layer.

7 **Gateway DoIP T_Data.con**: network layer issues to gateway application the confirmation of the completion of the response message.
Client T_Data.ind: network layer issues to diagnostic application the indication of the reception of the response message. Client reloads the P_{Client} timer with $P_{Client} = P6_{Client\_max}$.
Server #3 DoCAN T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{Server}$. Server #3 stops the $P2_{Server}$ timer.

8 **Server #3 DoCAN T_Data.con**: network layer issues to diagnostic application the confirmation of the completion of the response message.
Gateway DoCAN T_Data.ind: network layer issues to gateway application the indication of the reception of the response message.
Gateway DoIP_Data.req: gateway application forwards the response by issuing a T_data.req to the DoIP network layer.

9 **Gateway DoIP T_Data.con**: network layer issues to gateway application the confirmation of the completion of the response message.
Client T_Data.ind: network layer issues to diagnostic application the indication of the reception of the response message. Client reloads the P_{Client} timer with $P_{Client} = P6_{Client\_max}$.
Server #2 DoCAN T_Data.req: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{Server}$. Server #2 stops the $P2_{Server}$ timer.

10 **Server #2 DoCAN T_Data.con**: network layer issues to diagnostic application the confirmation of the completion of the response message.
Gateway DoCAN T_Data.ind: network layer issues to gateway application the indication of the reception of the response message.
Gateway DoIP_Data.req: gateway application forwards the response by issuing a T_data.req to the DoIP network layer.

11 **Gateway DoIP T_Data.con**: network layer issues to gateway application the confirmation of the completion of the response message.
Client T_Data.ind: network layer issues to diagnostic application the indication of the reception of the response message. Client reloads the P_{Client} timer with $P_{Client} = P6_{Client\_max}$. The P_{Client} timer expires when it reaches $P6_{Client\_max}$, indicating that no more responses are forthcoming.

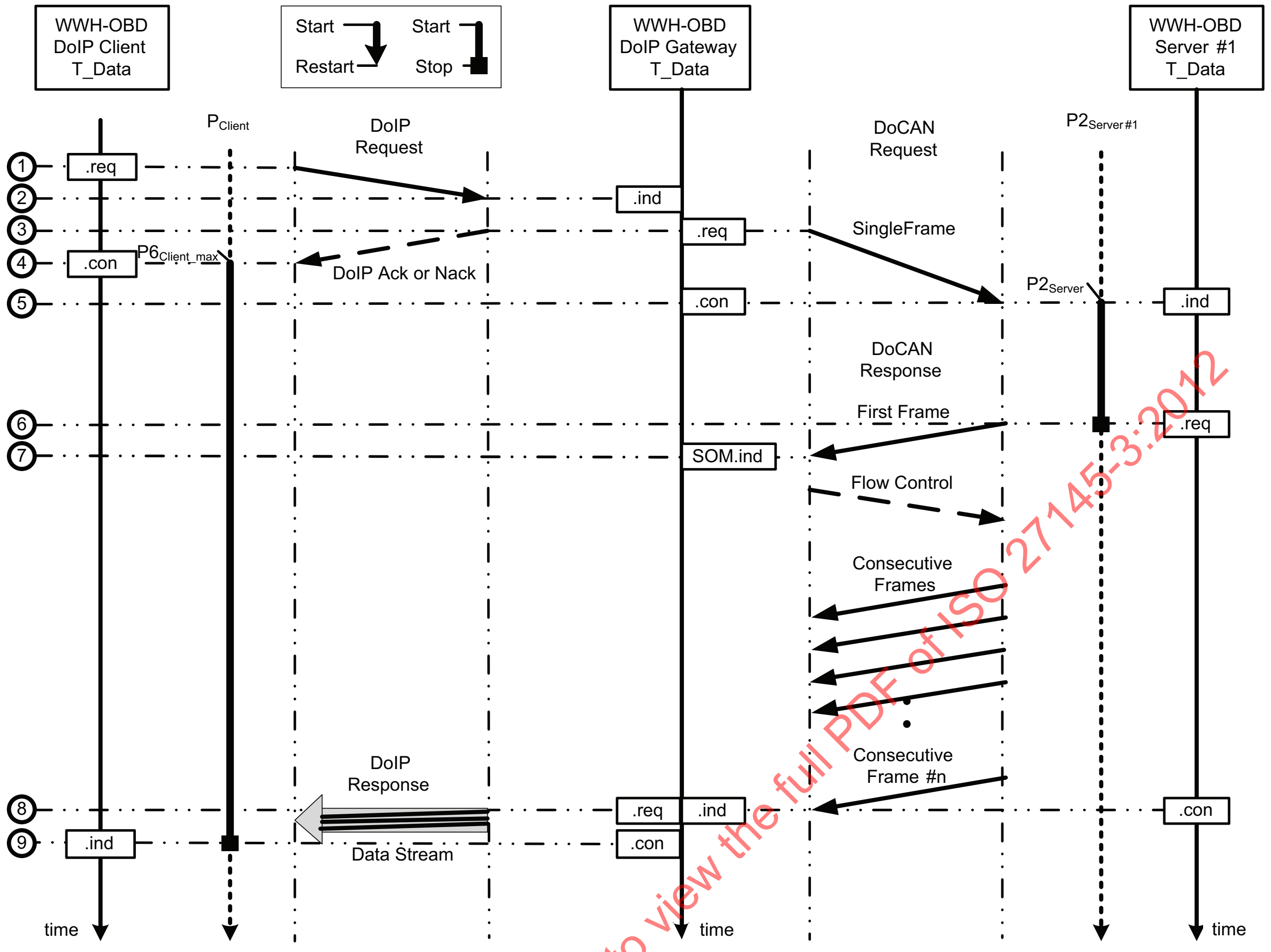
12 **Client T_Data.req**: diagnostic application issues a physically addressed request message to the DoIP network layer.

13 **Client T_Data.con**: network layer issues to diagnostic application the confirmation of the completion of the physically addressed request message. Client starts the P_{Client} timer with $P_{Client} = P6_{Client\_max}$.
Gateway DoIP T_Data.ind: network layer issues to gateway application the indication of the reception of a request message.
Gateway DoCAN T_Data.req: gateway application forwards the physically addressed request by issuing a T_Data.req to the DoCAN network layer.

14 **Gateway DoCAN T_Data.con**: network layer issues to gateway application the confirmation of the completion of the request message.
Servers #1 DoCAN T_Data.ind: network layer issues to diagnostic application the indication of the reception of a request message. Server #1 starts its $P2_{Server}$ timer using the default reload value $P2_{Server} = P2_{Server\ \_max}$.

Figure 9 — ISO 13400 DoIP timing diagram

Figure 10 illustrates a store-and-forward mechanism where it collects multiple frame responses in the DoIP gateway before it transmits a single response to the client.



Key

1 **Client T_Data.req**: diagnostic application issues a physically addressed request message to the network layer.

2 **Gateway DoIP T_Data.ind**: network layer issues to application the indication of the reception of a request message.

3 **Gateway DoIP**: application issues an Acknowledge/No Acknowledge (Ack/Nack) frame to the DoIP client.
Gateway DoCAN T_Data.req: application issues a physically addressed request message to the network layer.

4 **Client T_Data.con**: network layer issues to diagnostic application the reception of an acknowledge/no acknowledge from the DoIP Gateway. Client starts the P_{Client} timer with $P_{Client} = P6_{Client\_max}$.

5 **Server #1 DoCAN T_Data.ind**: network layer issues to diagnostic application the indication of the reception of a request message. Server #1 starts the $P2_{Server}$ timer using the value of $P2_{Server} = P2_{Server\_max}$.
Gateway DoCAN T_Data.con: network layer issues to application the confirmation of the completion of the request message.

6 **Server #1 DoCAN T_Data.req**: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{Server}$. Server #1 stops the $P2_{Server}$ timer.

7 **Gateway DoCAN T_DataSOM.ind**: network layer issues to application the indication of the reception of a StartOfMessage, which is initiated by the reception of a FirstFrame indication on CAN (see ISO 15765-2).

8 **Server #1 DoCAN T_Data.con**: network layer issues to diagnostic application the completion of the response message.
Gateway DoCAN T_Data.ind: network layer issues to application the indication of the completion of the response message.
Gateway DoIP T_Data.req: application has prepared the response message and issues a T_Data.req to network layer.

9 **Gateway DoIP T_Data.con**: network layer issues to application the completion of the response message.
Client T_Data.ind: network layer issues to diagnostic application the indication of the completion of the response message and stops the P_{Client} timer.

Figure 10 — ISO 13400 DoIP physically addressed short request with long response message

NOTE As TCP is a stream-oriented protocol, it is also possible that more than one DoIP response message is sent via a single TCP segment. Also, TCP uses acknowledging and automatic retries which depend on the overall network performance and reliability. Thus it is not possible to map individual responses into individual IP packets. Therefore, the message sequence charts (see Figure 9 and Figure 10) only provide a logical view of multiple messages on the Ethernet/IP/TCP side, which might differ from the actual IP packet transmission.

7.3.4 ISO 15765-4 DoCAN message timing definition

The message timing definition for the defaultSession shall be in accordance with Table 15. The values of the DoCAN-defined parameters shall be valid only for ISO 27145 WWH-OBD communication and services.

Table 15 — Message timing definition for ISO 15765-4 DoCAN in defaultSession

Timing parameter	Definition	Minimum [ms]	Maximum [ms]
$\Delta P2$ [a]	The $\Delta P2$ parameter is defined as the worst-case vehicle-network-design-dependent message transmission delay, such as delays introduced by gateways and bus-load arbitration delays. The value of $\Delta P2$ is divided between the time to transmit the request to the addressed server/ECU and the time to transmit the response to the client/tester.	> 0	50
$P2_{Server}$	The $P2_{Server}$ parameter is a performance requirement for the server/ECU to start with the response message after reception of a request message.	0	50
$P2_{Client}$	The $P2_{Client}$ parameter timeout is in order for the client to wait after the successful transmission of a request for the start of incoming response messages. $P2_{Client\_min} = P2_{Server\_max} + \Delta P2$	100	— [b]
$P2^*_{Server}$	The $P2^*_{Server}$ parameter is a performance requirement for the server to start with the response message after the transmission of a negative response message with the negative response code 0x78 (enhanced response timing).	0 [c]	5 000
$P2^*_{Client}$	The $P2^*_{Client}$ parameter is the enhanced timeout in order for the client to wait after reception of a negative response message with the negative response code (NRC) 0x78 for the start of incoming response messages. $P2^*_{Client\_min} = P2^*_{Server\_max} + \Delta P2_{max}$	5 050	— [d]
$P3_{Client\_Phys}$	The $P3_{Client\_Phys}$ parameter is the minimum time for the client to wait after successful transmission of a physically addressed request message with no response required before the next physically addressed request message can be transmitted. $P2_{Server\_max} + \Delta P2_{min}$	100	— [e]
$P3_{Client\_Func}$	The $P3_{Client\_Func}$ parameter is the minimum time for the client to wait after successful transmission of a functionally addressed request message before the next functionally addressed request message can be transmitted, in the case that no response is required or that the requested data is only supported by a subset of the functionally addressed servers. $P2_{Server\_max} + \Delta P2_{min}$	100	— [e]

[a] $\Delta P2$ is used for a worst-case consideration of in-vehicle delays. In DoCAN a fixed value is used for $\Delta P2$:
$\Delta P2 = \Delta P2_{Request} + \Delta P2_{Response} \leq 50$ ms
Clearly, $\Delta P2$ can be reduced to a value of 0 ms, i.e. a gateway will not delay responses to achieve a $\Delta P2$ of 50 ms!

[b] The maximum value a client uses for $P2_{Client}$ to wait for the start of a response message is left to the discretion of the client as long as it is greater than the specified minimum value of $P2_{Client}$. However, the vehicle has to start the response within the minimum value of $P2_{Client}$.

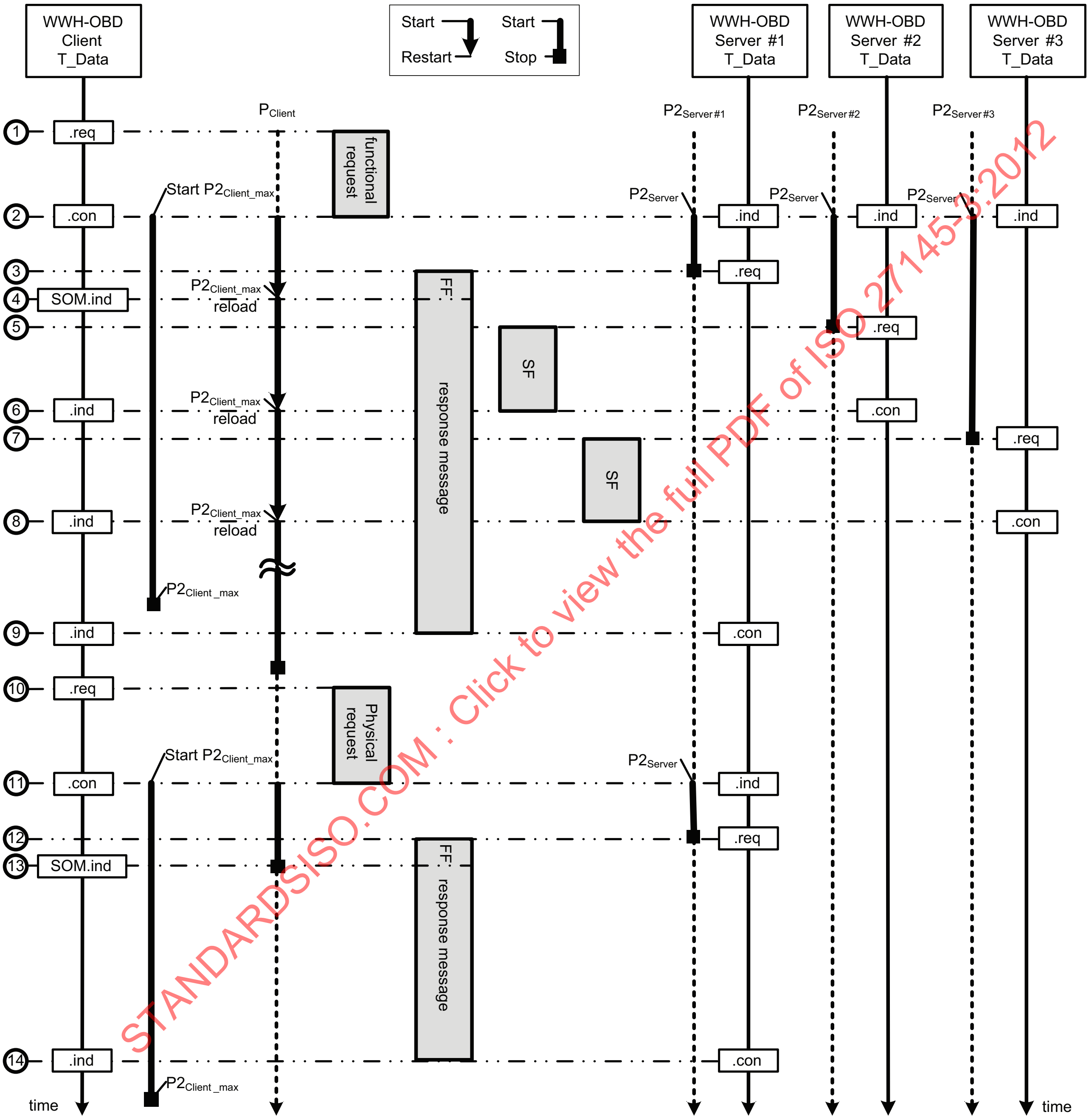
[c] During the enhanced response timing, the minimum time between the transmission of consecutive negative response messages, each with negative response code 0x78, shall be 0,3 × $P2^*_{Server\_max}$ in order to avoid flooding the data link with unnecessary negative response messages with response code 0x78.

[d] The value a client uses for $P2^*_{Client}$ to wait for the start of a response message is left to the discretion of the client as long as it is greater than the specified minimum value of $P2^*_{Client}$. However the vehicle has to respond within the minimum value of $P2_{Client}$.

[e] The maximum time a client waits until it transmits the next request message is at the discretion of the client.

IMPORTANT — For WWH-OBD the timing definitions in this part of ISO 27145 take precedence over other related documents.

Figure 11 illustrates a client functional request message with a multiple frame response from server #1 and single frame responses from servers #2 and #3.



Key

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

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

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

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

5 **Server #2 T_Data.req**: diagnostic application has prepared the response message and issues a T_Data.req to the network layer within $P2_{Server}$. The response message is a SingleFrame (SF) 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 P_{Client} with $P2_{Client\_max}$ value.

7 **Server #3 T_Data.req**: diagnostic application has prepared the response message and issues a T_Data.req to network layer within $P2_{Server}$. The response message is a SingleFrame (SF) 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 $P2_{Server}$ with $P2_{Server\_max}$ 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 physically addressed request message to network layer.

11 **Server #1 T_Data.ind**: network layer issues to diagnostic application the indication of the reception of a request message. Server #1 starts the $P2_{Server}$ timer using the value of $P2_{Server} = P2_{Server\_max}$.
Client T_Data.con: network layer issues to diagnostic application the confirmation of the completion of the request message. Client starts its P_{Client} timer using the default reload value $P2_{Client\_max}$.

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

13 **Client T_DataSOM.ind**: network layer issues to diagnostic application the indication of the reception of a StartOfMessage, which is initiated by the reception of a FirstFrame (FF) indication on DoCAN (see ISO 15765-2). Client stops P_{Client} because FirstFrame (FF) of response message is received from server #1.

14 **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 11 — ISO 15765-4 DoCAN timing diagram

8 Presentation layer requirements

The presentation layer shall be in accordance with ISO 27145-2.

9 Session layer requirements

The session layer shall be in accordance with ISO 14229-2.

Annex A
(informative)

WWH-OBD-related unified diagnostic service examples

A.1 ReadDataByIdentifier message flow examples

A.1.1 Read protocol identification InfoType identifier

The external test equipment uses functional addressing to request the SAE J1979-DA-defined ITID (0xF810) protocol identification.

This example shows the functionally addressed request message and the response of multiple WWH-OBD-compliant servers/ECUs providing the SAE J1979-DA-defined InfoType “protocol identification”:

— DID = 0xF810 contains the protocol identification.

Table A.1 defines the functionally addressed ReadDataByIdentifier request message to the vehicle.

Table A.1 — ReadDataByIdentifier functionally addressed request message flow example #1

Message direction:	client → vehicle		
Message type:	functionally addressed request message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier request SID	0x22	RDBI
#2	DataIdentifier #1 (HB) = ITID = Protocol Identification	0xF8	DID_HB
#3	DataIdentifier #1 (LB) = ITID = Protocol Identification	0x10	DID_LB

Each server that supports the DataIdentifier #1 sends a positive response message to the client.

Table A.2 (server #1) and Table A.3 (server #n) define the ReadDataByIdentifier positive response message.

Table A.2 — ReadDataByIdentifier positive response message flow example #1 — Server #1

Message direction:	server #1 → client		
Message type:	response message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier response SID	0x62	RDBIPR
#2	DataIdentifier #1 (HB) = ITID = PROTID	0xF8	DID_HB
#3	DataIdentifier #1 (LB) = ITID = PROTID	0x10	DID_LB
#4	ITID dataRecord[1].ITP_DB1; protocol identification =(0x01 = ISO 27145)	0x01	ITP_DB1

Table A.3 — ReadDataByIdentifier positive response message flow example #1 — Server #n

Message direction:	server #n → client		
Message type:	response message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier response SID	0x62	RDBIPR
#2	DataIdentifier #1 (HB) = ITID = PROTID	0xF8	DID_HB
#3	DataIdentifier #1 (LB) = ITID = PROTID	0x10	DID_LB
#4	ITID dataRecord[1].ITP_DB1; protocol identification =(0x01 = ISO 27145)	0x01	ITP_DB1

A.1.2 Read supported SAE J1979-DA ITIDs

The external test equipment requests from each server/ECU all supported SAE J1979-DA ITIDs. See ISO 27145-2:2012, Annex A for the definitions of supported PIDs/MIDs and ITIDs.

Table A.4 defines the ReadDataByIdentifier physically addressed request message flow example #2.

Table A.4 — ReadDataByIdentifier physically addressed request message flow example #2

Message direction:	client → server #1		
Message type:	physically addressed request message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier response SID	0x22	RDBI
#2	DataIdentifier #1 (HB) = ITID	0xF8	DID_HB
#3	DataIdentifier #1 (LB) = ITID	0x00	DID_LB

Table A.5 defines the ReadDataByIdentifier response message flow example #2 — Server #1.

Table A.5 — ReadDataByIdentifier response message flow example #2 — Server #1

Message direction:	server #1 → client		
Message type:	response message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier response SID	0x62	RDBI
#2	DataIdentifier #1 (HB) = MID	0xF8	DID_HB
#3	DataIdentifier #1 (LB) = MID	0x00	DID_LB
#4	ITID dataRecord[1].ITP_DB#1; ITIDs supported data byte #A	0x40	ITP_DB1
#5	ITID dataRecord[2].ITP_DB#2; ITIDs supported data byte #B	0x01	ITP_DB2
#6	ITID dataRecord[3].ITP_DB#3; ITIDs supported data byte #C	0x80	ITP_DB3
#7	ITID dataRecord[4].ITP_DB#4; ITIDs supported data byte #D	0x00	ITP_DB4

Server #1 (ECM) supports the following ITID:

— DID = 0xF802 contains the VIN;

— DID = 0xF810 contains the protocol identification;

— DID = 0xF811 contains the WWH-OBD GTR number.

As a result of the supported SAE J1979-DA ITID request, the external test equipment creates an internal list of supported DIDs for server #1.

The same request and response sequence applies to all servers which previously sent a positive response to the functionally addressed request message with the SAE J1979-DA-defined ITID (0xF810) protocol identification (ISO 27145 supported).

A.1.3 Read single SAE J1979-DA InfoType identifier

This example demonstrates the request and response message sequence to retrieve the VIN.

Table A.6 defines the functionally addressed ReadDataByIdentifier request message flow example #3.

Table A.6 — ReadDataByIdentifier functionally addressed request message flow example #3 — Vehicle

Message direction:	client → vehicle		
Message type:	functionally addressed request message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier request SID	0x22	RDBI
#2	DataIdentifier #1 (HB) = ITID = VIN	0xF8	DID_HB
#3	DataIdentifier #1 (LB) = ITID = VIN	0x02	DID_LB

Table A.7 defines the ReadDataByIdentifier positive response message from one WWH-OBD GTR-compliant server.

Table A.7 — ReadDataByIdentifier positive response message flow example #3 — Server #1

Message direction:	server #1 → client		
Message type:	response message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier response SID	0x62	RDBIPR
#2	DataIdentifier #1 (HB) = ITID = VIN	0xF8	DID_HB
#3	DataIdentifier #1 (LB) = ITID = VIN	0x02	DID_LB
#4	ITID dataRecord[1].ITP_DB1; VIN Digit 1 = “W”	0x57	ITP_DB1
#5	ITID dataRecord[2].ITP_DB2; VIN Digit 2 = “0”	0x30	ITP_DB2
#6	ITID dataRecord[3].ITP_DB3; VIN Digit 3 = “L”	0x4C	ITP_DB3
#7	ITID dataRecord[4].ITP_DB4; VIN Digit 4 = “0”	0x30	ITP_DB4
#8	ITID dataRecord[5].ITP_DB5; VIN Digit 5 = “0”	0x30	ITP_DB5
#9	ITID dataRecord[6].ITP_DB6; VIN Digit 6 = “0”	0x30	ITP_DB6
#10	ITID dataRecord[7].ITP_DB7; VIN Digit 7 = “0”	0x30	ITP_DB7
#11	ITID dataRecord[8].ITP_DB8; VIN Digit 8 = “4”	0x34	ITP_DB8
#12	ITID dataRecord[9].ITP_DB9; VIN Digit 9 = “3”	0x33	ITP_DB9
#13	ITID dataRecord[10].ITP_DB10; VIN Digit 10 = “M”	0x4D	ITP_DB10
#14	ITID dataRecord[11].ITP_DB11; VIN Digit 11 = “B”	0x42	ITP_DB11
#15	ITID dataRecord[12].ITP_DB12; VIN Digit 12 = “5”	0x35	ITP_DB12
#16	ITID dataRecord[13].ITP_DB13; VIN Digit 13 = “4”	0x34	ITP_DB13
#17	ITID dataRecord[14].ITP_DB14; VIN Digit 14 = “1”	0x31	ITP_DB14
#18	ITID dataRecord[15].ITP_DB15; VIN Digit 15 = “3”	0x33	ITP_DB15
#19	ITID dataRecord[16].ITP_DB16; VIN Digit 16 = “2”	0x32	ITP_DB16
#20	ITID dataRecord[17].ITP_DB17; VIN Digit 17 = “6”	0x36	ITP_DB17

A.1.4 Read supported SAE J1979-DA DIDs

The external test equipment requests from each server/ECU all supported SAE J1979-DA PIDs. See ISO 27145-2:2012, Annex A for the definition of supported PIDs/MIDs and ITIDs.

Table A.8 — ReadDataByIdentifier physically addressed request message flow example #4 — Server #1

Message direction:	client → server #1		
Message type:	physically addressed request message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier response SID	0x22	RDBI
#2	DataIdentifier #1 (HB) = PID	0xF4	DID_HB
#3	DataIdentifier #1 (LB) = PID	0x00	DID_LB

Table A.9 — ReadDataByIdentifier response message flow example #4 — Server #1

Message direction:	server #1 → client		
Message type:	response message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier request SID	0x62	RDBI
#2	DataIdentifier #1 (HB) = PID	0xF4	DID_HB
#3	DataIdentifier #1 (LB) = PID	0x00	DID_LB
#4	PID dataRecord[1].DP_DB#1; PIDs supported data byte #A	0x10	DP_DB1
#5	PID dataRecord[2].DP_DB#2; PIDs supported data byte #B	0x10	DP_DB2
#6	PID dataRecord[3].DP_DB#3; PIDs supported data byte #C	0x00	DP_DB3
#7	PID dataRecord[4].DP_DB#4; PIDs supported data byte #D	0x00	DP_DB4

Server #1 (ECM) supports the following DIDs:

— DID = 0xF404 contains Calculated Load Value;

— DID = 0xF40C contains Engine RPM.

As a result of the supported SAE J1979-DA PID request, the external test equipment creates an internal list of supported DIDs for server #1.

Table A.10 — ReadDataByIdentifier physically addressed request message flow example #4 — Server #2

Message direction:	client → server #2		
Message type:	physically addressed request message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier response SID	0x22	RDBI
#2	DataIdentifier #1 (HB) = PID	0xF4	DID_HB
#3	DataIdentifier #1 (LB) = PID	0x00	DID_LB

Table A.11 — ReadDataByIdentifier response message flow example #4 — Server #2

Message direction:	server #2 → client		
Message type:	response message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier request SID	0x62	RDBI
#2	DataIdentifier #1 (HB) = PID	0xF4	DID_HB
#3	DataIdentifier #1 (LB) = PID	0x00	DID_LB
#4	PID dataRecord[1].DP_DB#1; PIDs supported data byte #A	0x00	DP_DB1
#5	PID dataRecord[2].DP_DB#2; PIDs supported data byte #B	0x08	DP_DB2
#6	PID dataRecord[3].DP_DB#3; PIDs supported data byte #C	0x00	DP_DB3
#7	PID dataRecord[4].DP_DB#4; PIDs supported data byte #D	0x00	DP_DB4

Server #2 (TCM) supports the following DID:

— DID = 0xF40D contains Vehicle Speed Sensor.

As a result of the supported SAE J1979-DA PID request, the external test equipment creates an internal list of supported DIDs for server #2.

The same request and response sequence applies to all servers which previously sent a positive response to the functionally addressed request message with the SAE J1979-DA-defined ITID (0xF810) protocol identification (ISO 27145 supported).

A.1.5 Read multiple SAE J1979-DA-defined PIDs

The external test equipment requests from each server/ECU all supported SAE J1979-DA PIDs (0xF400). See ISO 27145-2:2012, Annex A for the definition of supported PIDs/MIDs and ITIDs.

Table A.12 defines the physically addressed ReadDataByIdentifier request to server #1 message flow example #5.

Table A.12 — ReadDataByIdentifier physically addressed request message flow example #5 — Server #1

Message direction:	client → server #1		
Message type:	physically addressed request message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier request SID	0x22	RDBI
#2	DataIdentifier #1 (HB) = PID = Calculated Load Value	0xF4	DID_HB
#3	DataIdentifier #1 (LB) = PID = Calculated Load Value	0x04	DID_LB
#4	DataIdentifier #2 (HB) = PID = Engine RPM	0xF4	DID_HB
#5	DataIdentifier #2 (LB) = PID = Engine RPM	0x0C	DID_LB

Table A.13 defines the ReadDataByIdentifier positive response from sever #1 message flow example #5.

Table A.13 — ReadDataByIdentifier positive response message flow example #5 — From server #1

Message direction:	server #1 → client		
Message type:	response message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier response SID	0x62	RDBIPR
#2	DataIdentifier #1 (HB) = PID = Calculated Load Value	0xF4	DID_HB
#3	DataIdentifier #1 (LB) = PID = Calculated Load Value	0x04	DID_LB
#4	PID dataRecord[1].DP_DB#1; Calculated Load Value = 13,7 %	0x23	DP_DB1
#5	DataIdentifier #2 (HB) = PID = Engine RPM	0xF4	DID_HB
#6	DataIdentifier #2 (LB) = PID = Engine RPM	0x0C	DID_LB
#7	PID dataRecord[1].DP_DB#1; Engine RPM = 1500 min^{-1}	0x17	DP_DB1
#8	PID dataRecord[2].DP_DB#2; Engine RPM = 1500 min^{-1}	0x70	DP_DB2

Table A.14 defines the ReadDataByIdentifier physically addressed request to server #2 message flow example #6.

Table A.14 — ReadDataByIdentifier physically addressed request message flow example #6 — Server #2

Message direction:	client → server #2		
Message type:	physically addressed request message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier request SID	0x22	RDBI
#2	DataIdentifier #3 (HB) = PID = Vehicle Speed Sensor	0xF4	DID_HB
#3	DataIdentifier #3 (LB) = PID = Vehicle Speed Sensor	0x0D	DID_LB

Table A.15 defines the ReadDataByIdentifier positive response from server #2 message flow example #6.

Table A.15 — ReadDataByIdentifier positive response message flow example #6 — Server #2

Message direction:	server#2 → client		
Message type:	response message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier response SID	0x62	RDBIPR
#2	DataIdentifier #3 (HB) = PID = Vehicle Speed Sensor	0xF4	DID_HB
#3	DataIdentifier #3 (LB) = PID = Vehicle Speed Sensor	0x0D	DID_LB
#4	PID_DataRecord[1].DP_DB#1; Vehicle Speed Sensor = 0 km/h	0x00	DP_DB1

A.1.6 Read single SAE J1979-DA monitor identifier

A.1.6.1 Step #1: Request supported SAE J1979-DA monitor identifiers

The external test equipment requests from each server/ECU all supported SAE J1979-DA MIDs (0xF600). See ISO 27145-2:2012, Annex A for the definition of supported PIDs/MIDs and ITIDs.

Table A.16 — ReadDataByIdentifier physically addressed request message flow example #7 — Server #1

Message direction:	client → server #1		
Message type:	physically addressed request message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier request SID	0x22	RDBI
#2	DataIdentifier #1 (HB) = MID	0xF6	DID_HB
#3	DataIdentifier #1 (LB) = MID	0x00	DID_LB

Table A.17 — ReadDataByIdentifier response message flow example #7 — Server #1

Message direction:	server #1 → client		
Message type:	response message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier request SID	0x62	RDBI
#2	DataIdentifier #1 (HB) = MID	0xF6	DID_HB
#3	DataIdentifier #1 (LB) = MID	0x00	DID_LB
#4	MID dataRecord[1].MP_DB#1; MIDs supported data byte #A	0xCC	MP_DB1
#5	MID dataRecord[2].MP_DB#2; MIDs supported data byte #B	0x00	MP_DB2
#6	MID dataRecord[3].MP_DB#3; MIDs supported data byte #C	0x00	MP_DB3
#7	MID dataRecord[4].MP_DB#4; MIDs supported data byte #D	0x01	MP_DB4

MID 0xF620 is supported because data byte #D.0 = “1”. In order to identify additional supported MIDs, another request with MID 0xF620 needs to be sent to server #1.

Table A.18 — ReadDataByIdentifier physically addressed request message flow example #7 — Server #1

Message direction:	client → server #1		
Message type:	physically addressed request message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier request SID	0x22	RDBI
#2	DataIdentifier #1 (HB) = MID	0xF6	DID_HB
#3	DataIdentifier #1 (LB) = MID	0x20	DID_LB

Table A.19 — ReadDataByIdentifier response message flow example #7 — Server #1

Message direction:	server #1 → client		
Message type:	response message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier request SID	0x62	RDBI
#2	DataIdentifier #1 (HB) = MID	0xF6	DID_HB
#3	DataIdentifier #1 (LB) = MID	0x20	DID_LB
#4	MID dataRecord[1].MP_DB#1; MIDs supported data byte #A	0xC0	MP_DB1
#5	MID dataRecord[2].MP_DB#2; MIDs supported data byte #B	0x00	MP_DB2
#6	MID dataRecord[3].MP_DB#3; MIDs supported data byte #C	0x00	MP_DB3
#7	MID dataRecord[4].MP_DB#4; MIDs supported data byte #D	0x00	MP_DB4

MID 0xF640 is not supported because data byte #D.0 = "0". This indicates that no additional MIDs are supported.

As a result of the supported SAE J1979-DA MID request, the external test equipment creates an internal list of supported MIDs for each server/ECU.

Server #1 (ECM) supports the following OBD monitor identifiers:

— 0xF601: Oxygen Sensor Monitor Bank 1 – Sensor 1;

— 0xF602: Oxygen Sensor Monitor Bank 1 – Sensor 2;

— 0xF605: Oxygen Sensor Monitor Bank 2 – Sensor 1;

— 0xF606: Oxygen Sensor Monitor Bank 2 – Sensor 2;

— 0xF621: Catalyst Monitor Bank 1;

— 0xF622: Catalyst Monitor Bank 2.

Server #2 (TCM) does not support any OBD monitor identifiers.

The same request and response sequence applies to all servers which previously sent a positive response to the functionally addressed request message with the SAE J1979-DA-defined ITID (0xF810) protocol identification (ISO 27145 supported).

A.1.6.2 Step #2: Request current powertrain diagnostic data

Prior to requesting OBD monitor test results, the external test equipment shall evaluate whether the monitor is complete. The status of the monitor is included in the response message of the service ReadDataByIdentifier, PID 0xF401 data bytes.

A.1.6.3 Step #3: Request on-board monitoring test results for specific monitored systems

The external test equipment sends a physically addressed ReadDataByIdentifier request message to read on-board monitoring test results for specific monitored systems with one OBD monitor identifier in the request message to server/ECU #1.

This example demonstrates the request and response of a single OBD monitor data identifier where MID = 0xF601 contains the Oxygen Sensor Monitor Bank 1 – Sensor 1 with three different OBD test IDs: 0x01, 0x05 and 0x85.

Table A.20 defines the physically addressed ReadDataByIdentifier request to server #1 message flow example #8.

Table A.20 — ReadDataByIdentifier physically addressed requestmessage flow example #8 — Server #1

Message direction:	client → server#1		
Message type:	physically addressed request message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier request SID	0x22	RDBI
#2	DataIdentifier#1 (HB) = MID = Oxygen Sensor Monitor B1 – S1	0xF6	DID_HB
#3	DataIdentifier#1 (LB) = MID = Oxygen Sensor Monitor B1 – S1	0x01	DID_LB

Table A.21 defines the ReadDataByIdentifier positive response from server #1 message flow example #8.

Table A.21 — ReadDataByIdentifier positive response message flow example #8 — Server #1

Message direction:	server #1 → client		
Message type:	response message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier response SID	0x62	RDBIPR
#2	DataIdentifier#1 (HB) = MID = Oxygen Sensor Monitor B1 – S1	0xF6	DID_HB
#3	DataIdentifier#1 (LB) = MID = Oxygen Sensor Monitor B1 – S1	0x01	DID_LB
#4	MID dataRecord[1].MP_SMTID; Test ID: Rich to lean sensor threshold voltage (constant)	0x01	MP_SMTID
#5	MID dataRecord[2].MP_UASID; Unit and Scaling ID: Voltage	0x0A	MP_UASID
#6	MID dataRecord[3].MP_TVHI; Test Value 0,365 V	0x0B	MP_TVHI
#7	MID dataRecord[4].MP_TVLO; Test Value 0,365 V	0xB0	MP_TVLO
#8	MID dataRecord[5].MP_MINTLHI; Min. Test Limit 0,365 V	0x0B	MP_MINTLHI
#9	MID dataRecord[6].MP_MINTLLO; Min. Test Limit 0,365 V	0xB0	MP_MINTHLO
#10	MID dataRecord[7].MP_MAXTLHI; Max. Test Limit 0,365 V	0x0B	MP_MAXTLHI
#11	MID dataRecord[8].MP_MAXTLLO; Max. Test Limit 0,365 V	0xB0	MP_MAXTLLO
#12	MID dataRecord[1].MP_SMTID; Test ID: Rich to lean sensor switch time (calculated)	0x05	MP_SMTID
#13	MID dataRecord[2].MP_UASID; Unit and Scaling ID: Time	0x10	MP_UASID
#14	MID dataRecord[3].MP_TVHI; Test Value (HB) 0,072 s	0x00	MP_TVHI
#15	MID dataRecord[4].MP_TVLO; Test Value (LB) 0,072 s	0x48	MP_TVLO
#16	MID dataRecord[5].MP_MINTLHI; Min. Test Limit 0,000 s	0x00	MP_MINTLHI
#17	MID dataRecord[6].MP_MINTLLO; Min. Test Limit 0,000 s	0x00	MP_MINTHLO
#18	MID dataRecord[7].MP_MAXTLHI; Max. Test Limit 0,100 s	0x00	MP_MAXTLHI
#19	MID dataRecord[8].MP_MAXTLLO; Max. Test Limit 0,100 s	0x64	MP_MAXTLLO
#20	MID dataRecord[1].MP_SMTID; Test ID: the name of this Test ID shall be documented in the vehicle Service Information!	0x85	MP_SMTID
#21	MID dataRecord[2].MP_UASID; Unit and Scaling ID: Counts	0x24	MP_UASID
#22	MID dataRecord[3].MP_TVHI; Test Value 150 counts	0x00	MP_TVHI
#23	MID dataRecord[4].MP_TVLO: Test Value 150 counts	0x96	MP_TVLO
#24	MID dataRecord[5].MP_MINTLHI: Min. Test Limit 75 counts	0x00	MP_MINTLHI
#25	MID dataRecord[6].MP_MINTLLO: Min. Test Limit 75 counts	0x4B	MP_MINTLLO
#26	MID dataRecord[7].MP_MAXTLHI: Max. Test Limit 65 535 counts	0xFF	MP_MAXTLHI
#27	MID dataRecord[8].MP_MAXTLLO: Max. Test Limit 65 535 counts	0xFF	MP_MAXTLLO

A.1.6.4 Step #4: Request on-board monitoring test results for specific monitored systems

In this example the requested monitor has not been completed once. This example demonstrates the request and response of a single OBD monitor data identifier where:

— MID = 0xF621 reports one Test ID: 0x87;

— MID = 0xF621 contains the Catalyst Monitor Bank 1 test results for Test ID 0x87.

Table A.22 defines the physically addressed ReadDataByIdentifier request message to server #1 message flow example #9.

Table A.22 — ReadDataByIdentifier physically addressed request message flow example #9 — Server #1

Message direction:	client → server #1		
Message type:	physically addressed request message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier request SID	0x22	RDBI
#2	DataIdentifier#1 (HB) = MID = Catalyst Monitor Bank 1	0xF6	DID_HB
#3	DataIdentifier#1 (LB) = MID = Catalyst Monitor Bank 1	0x21	DID_LB

Table A.23 defines the ReadDataByIdentifier positive response message from server #1 message flow example #9.

Table A.23 — ReadDataByIdentifier positive response message flow example #9 — Server #1

Message direction:	server #1 → client		
Message Type:	response message		
A_Data byte	**Description**	**Byte value**	**Mnemonic**
#1	ReadDataByIdentifier response SID	0x62	RDBIPR
#2	DataIdentifier#1 (HB) = MID = Catalyst Monitor Bank 1	0xF6	DID_HB
#3	DataIdentifier#1 (LB) = MID = Catalyst Monitor Bank 1	0x21	DID_LB
#4	MID dataRecord[1].MP_SMTID; Test ID: the name of this Test ID shall be documented in the vehicle Service Information!	0x87	MP_SMTID
#5	MID dataRecord[2].MP_UASID; Unit and Scaling ID: Percent	0x2E	MP_UASID
#6	MID dataRecord[3].MP_TVHI; Test Value 0,00 %	0x00	MP_TVHI
#7	MID dataRecord[4].MP_TVLO: Test Value 0,00 %	0x00	MP_TVLO
#8	MID dataRecord[5].MP_MINTLHI: Min. Test Limit 0,00 %	0x00	MP_MINTLHI
#9	MID dataRecord[6].MP_MINTLLO: Min. Test Limit 0,00 %	0x00	MP_MINTLLO
#10	MID dataRecord[7].MP_MAXTLHI: Max. Test Limit 0,00 %	0x00	MP_MAXTLHI
#11	MID dataRecord[8].MP_MAXTLLO: Max. Test Limit 0,00 %	0x00	MP_MAXTLLO

A.2 RoutineControl message flow examples

A.2.1 General assumption

For all examples, the client requests a response message by setting the suppressPosRspMsgIndicationBit (bit 7 of the sub-function parameter) to “FALSE” (“0”).