

First edition  
2012-12-01

Corrected version  
2013-01-15

---

---

**Road vehicles — Unified diagnostic  
services (UDS) —**

**Part 3:  
Unified diagnostic services on CAN  
implementation (UDSonCAN)**

*Véhicules routiers — Services de diagnostic unifiés (SDU) —*

*Partie 3: SDU sur l'implémentation du gestionnaire de réseau de  
communication (SDU sur CAN)*



Reference number  
ISO 14229-3:2012(E)

© ISO 2012



**COPYRIGHT PROTECTED DOCUMENT**

© ISO 2012

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Case postale 56 • CH-1211 Geneva 20  
Tel. + 41 22 749 01 11  
Fax + 41 22 749 09 47  
E-mail [copyright@iso.org](mailto:copyright@iso.org)  
Web [www.iso.org](http://www.iso.org)

Published in Switzerland

# Contents

Page

<b>Foreword</b>	<b>iv</b>
<b>Introduction</b>	<b>v</b>
<b>1 Scope</b>	<b>1</b>
<b>2 Normative references</b>	<b>1</b>
<b>3 Terms, definitions and abbreviated terms</b>	<b>1</b>
3.1 Terms and definitions	1
3.2 Abbreviated terms	1
<b>4 Conventions</b>	<b>2</b>
<b>5 Document overview</b>	<b>2</b>
<b>6 Unified diagnostic services implementation on CAN</b>	<b>4</b>
6.1 General	4
6.2 UDS on CAN services overview	4
6.3 ResponseOnEvent (0x86) service DoCAN implementation requirements	5
6.4 ReadDataByPeriodicIdentifier (0x2A) service	7
<b>7 Application layer requirements</b>	<b>13</b>
7.1 Application layer services	13
7.2 Application layer protocol	13
7.3 Application layer timing	13
<b>8 Presentation layer requirements</b>	<b>13</b>
<b>9 Session layer requirements</b>	<b>13</b>
<b>10 Transport/Network layer interface adaptation</b>	<b>13</b>
10.1 General information	13
10.2 DoCAN Transport/Network layer interface adaptation	14
<b>11 Data link layer diagnostic implementation requirements</b>	<b>15</b>
11.1 Usage of legislated 11-bit OBD CAN Identifiers for enhanced diagnostics	15
11.2 Usage of legislated 29-bit OBD CAN Identifiers for enhanced diagnostics	15
<b>Bibliography</b>	<b>16</b>

## 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 14229-3 was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 3, *Electrical and electronic equipment*.

This first edition of ISO 14229-3 cancels and replaces ISO 15765-3:2004.

This corrected version incorporates the above information regarding ISO 14229-3 cancelling and replacing ISO 15765-3:2004.

ISO 14229 consists of the following parts, under the general title *Road vehicles — Unified diagnostic services (UDS)*:

- *Part 1: Specification and requirements*
- *Part 2: Session layer services*
- *Part 3: Unified diagnostic services on CAN implementation (UDSonCAN)*
- *Part 4: Unified diagnostic services on FlexRay implementation (UDSonFR)*
- *Part 5: Unified diagnostic services on Internet Protocol implementation (UDSonIP)*
- *Part 6: Unified diagnostic services on K-Line implementation (UDSonK-Line)*

The following parts are under preparation / are planned:

- *Part 7: Unified diagnostic services on Local Interconnect Network implementation (UDSonLIN)*

## Introduction

This part of ISO 14229 has been established in order to enable the implementation of unified diagnostic services, as specified in ISO 14229-3, on Controller Area Networks (UDSonCAN).

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 14229 are divided into

- Application layer (layer 7):
  - Vehicle manufacturer enhanced diagnostics: ISO 14229-1, ISO 14229-3,
  - Legislated OBD: ISO 15031-5,
  - Legislated WWH-OBD: ISO 14229-1 / ISO 27145-3;
- Presentation layer (layer 6):
  - Vehicle manufacturer enhanced diagnostics: vehicle manufacturer specific,
  - Legislated OBD: SAE J1930-DA, SAE J1979-DA, SAE J2012-DA,
  - Legislated WWH-OBD: 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):
  - Vehicle manufacturer enhanced diagnostics: ISO 14229-2,
  - Legislated OBD: ISO 14229-2,
  - Legislated WWH-OBD: ISO 14229-2;
- Transport layer services (layer 4):
  - Vehicle manufacturer enhanced diagnostics: ISO 15765-2,
  - Legislated OBD: ISO 15765-2, ISO 15765-4
  - Legislated WWH-OBD: ISO 27145-4;
- Network layer services (layer 3):
  - Vehicle manufacturer enhanced diagnostics: ISO 15765-2,
  - Legislated OBD: ISO 15765-2, ISO 15765-4
  - Legislated WWH-OBD: ISO 27145-4;
- Data link layer (layer 2):
  - Vehicle manufacturer enhanced diagnostics: ISO 11898-1, ISO 11898-2, ISO 11898-3, ISO 11898-5,
  - Legislated OBD: ISO 11898-1, ISO 11898-2, ISO 15765-4,
  - Legislated WWH-OBD: ISO 27145-4;
- Physical layer (layer 1):
  - Vehicle manufacturer enhanced diagnostics: ISO 11898-1, ISO 11898-2, ISO 11898-3, ISO 11898-5,
  - Legislated OBD: ISO 11898-1, ISO 11898-2, ISO 15765-4,

— Legislated WWH-OBD: ISO 27145-4;

in accordance with Table 1.

**Table 1 — DoCAN enhanced diagnostics, legislated OBD and WWH-OBD specification reference applicable to the OSI layers**

Applicability	OSI 7 layers	Vehicle manufacturer enhanced diagnostics	Legislated OBD (On-Board Diagnostics)	Legislated WWH-OBDD (On-Board Diagnostics)		
Seven layer according to ISO/IEC 7498-1 and ISO/IEC 10731	Application (layer 7)	ISO 14229-1, ISO/FDIS 14229-3	ISO 15031-5	ISO 14229-1, ISO 27145-3		
	Presentation (layer 6)	vehicle manufacturer specific	SAE J1930-DA, SAE J1979-DA, SAE J2012-DA	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	ISO 15765-2, ISO 15765-4	ISO 15765-2, ISO 15765-4	ISO 27145-4	ISO 13400-2
	Network (layer 3)					
	Data link (layer 2)	ISO 11898-1, ISO 11898-2, ISO 11898-3, ISO 11898-5	ISO 11898-1, ISO 11898-2, ISO 15765-4	ISO 11898-1, ISO 11898-2, ISO 15765-4		ISO 13400-3, IEEE 802.3
	Physical (layer 1)					

# Road vehicles — Unified diagnostic services (UDS) —

## Part 3:

## Unified diagnostic services on CAN implementation (UDSonCAN)

### 1 Scope

This part of ISO 14229 specifies the implementation of a common set of unified diagnostic services (UDS) on controller area networks (CAN) in road vehicles (UDSonCAN).

UDSonCAN references ISO 14229-1 and ISO 14229-2 and specifies implementation requirements of the diagnostic services to be used for diagnostic communication over CAN.

NOTE UDSonCAN does not specify any requirement for the in-vehicle CAN bus architecture.

This part of ISO 14229 does not include any redundant information of the documents as listed in the Introduction. It focuses on

- additional requirements specific to the implementation of UDS on the CAN network, and
- specific restrictions in the implementation of UDS on the CAN network.

### 2 Normative references

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

ISO 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 services*

ISO 15765-1, *Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) — Part 1: General information and use case definition*

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

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

### 3 Terms, definitions and abbreviated terms

#### 3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 14229-1, ISO 14229-2, ISO 15765-1, ISO 15765-2 and ISO 15765-4 apply.

### **3.2 Abbreviated terms**

CF	consecutive frame
DA	destination address
DLC	data length code
FF	first frame
FC	flow control
ID	identifier
NA	network address
SA	source address
SF	single frame
SM	subnet mask
SOM	start of message
STRT	serviceToRespondTo
TA	target address
UDS	unified diagnostic services
USDT	unacknowledged segmented data transfer
UUDT	unacknowledged unsegmented data transfer

## **4 Conventions**

This part of ISO 14229 is based on the conventions discussed in the OSI Service Conventions (ISO/IEC 10731:1994) as they apply for diagnostic services.

## **5 Document overview**

Figure 1 illustrates the documents required to implement UDSONCAN.



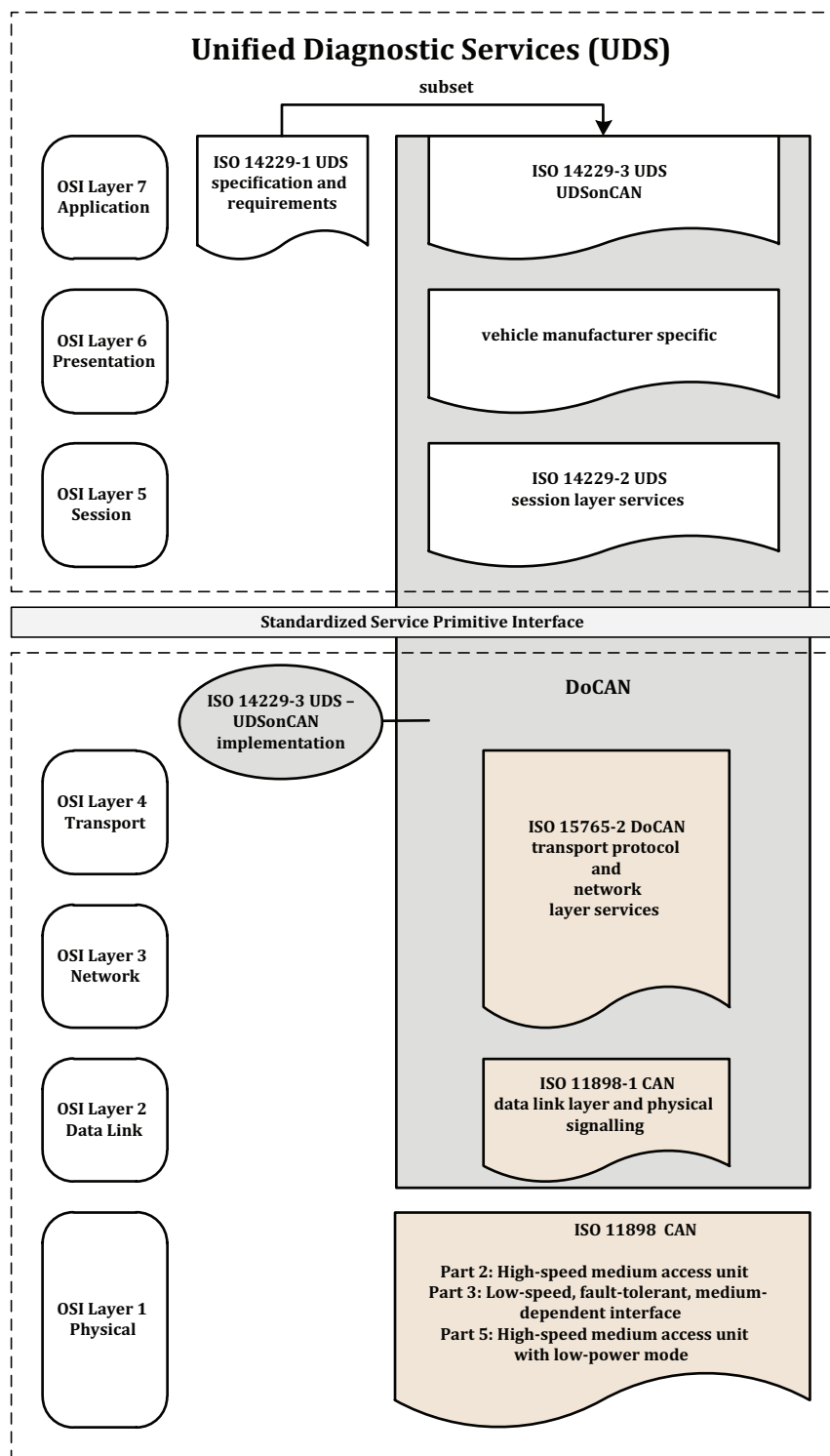


Figure 1 — UDSONCAN document reference according to OSI model

## 6 Unified diagnostic services implementation on CAN

### 6.1 General

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

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 CAN services overview

The purpose of Table 2 is to reference all ISO 14229-1 and ISO 14229-2 services as they are applicable for an implementation in ISO 14229-3, UDSONCAN. Table 2 contains the sum of all applicable services. Certain applications using this part of ISO 14229 to implement UDSONCAN may restrict the number of useable services and may categorize them in certain application areas/diagnostic sessions (default session, programming session, etc.).

Services in Table 2 that are marked “No CAN specific requirements” shall be implemented as defined in ISO 14229-1 and ISO 14229-2 with no additional restrictions. Services that are marked “CAN specific requirements” shall be implemented as defined by the subclause listed in the Table 2 entry.

**Table 2 — Overview of applicable ISO 14229-1, Unified diagnostic services and data ranges**

Diagnostic service name (ISO 14229-1)	Comment	Reference in this document
<b>Diagnostic and Communication Management Functional Unit</b>		
DiagnosticSessionControl	No CAN specific requirements	—
ECUReset	No CAN specific requirements	—
SecurityAccess	No CAN specific requirements	—
CommunicationControl	No CAN specific requirements	—
TesterPresent	No CAN specific requirements	—
SecuredData-Transmission	No CAN specific requirements	—
ControlDTCSetting	No CAN specific requirements	—
ResponseOnEvent	CAN specific requirements	see 6.3
LinkControl	No CAN specific requirements	—
<b>Data Transmission Functional Unit</b>		
ReadDataByIdentifier	No CAN specific requirements	—
ReadMemoryByAddress	No CAN specific requirements	—
ReadScalingDataByIdentifier	No CAN specific requirements	—
ReadDataByPeriodicIdentifier	CAN specific requirements	see 6.4
DynamicallyDefineDataIdentifier	No CAN specific requirements	—
WriteDataByIdentifier	No CAN specific requirements	—
WriteMemoryByAddress	No CAN specific requirements	—
ReadDTCInformation	No CAN specific requirements	—
ClearDiagnosticInformation	No CAN specific requirements	—
<b>Data Transmission Functional Unit</b>		
ReadDTCInformation	No CAN specific requirements	—
ClearDiagnosticInformation	No CAN specific requirements	—

Table 2 (continued)

Input/Output Control Functional Unit		
InputOutputControlByIdentifier	No CAN specific requirements	—
Remote Activation Of Routine Functional Unit		
RoutineControl	No CAN specific requirements	—
Upload/Download Functional Unit		
RequestDownload	No CAN specific requirements	—
RequestUpload	No CAN specific requirements	—
TransferData	No CAN specific requirements	—
RequestTransferExit	No CAN specific requirements	—
RequestFileTransfer	No CAN specific requirements	—

### 6.3 ResponseOnEvent (0x86) service DoCAN implementation requirements

In addition to the generic implementation requirements stated in ISO 14229-1 the following additional CAN specific implementation requirements shall apply.

While the ResponseOnEvent service is active, the server shall be able to process concurrent diagnostic request and response messages accordingly. This should be accomplished with a different serviceToRespondTo CAN Identifiers. If the same diagnostic request/response CAN Identifiers are used for diagnostic communication and the serviceToRespondTo-responses, the following restrictions shall apply:

- a) The server may ignore an incoming diagnostic request issued by the client that initiated the event, after an event has occurred and the serviceToRespondTo-response is in progress, until the serviceToRespondTo-response is completed. If another client issued the request, the server may or may not ignore the request, depending on the server's capabilities.
- b) When the client receives any response after sending a diagnostic request, the response shall be classified according to the possible serviceToRespondTo-responses and the expected diagnostic responses to the previously sent request.
  - 1) If the response is a serviceToRespondTo-response (one of the possible responses set up with ResponseOnEvent-service), the client shall repeat the request after the serviceToRespondTo-response has been received completely.
  - 2) If the response is ambiguous (i.e. the response could originate from the serviceToRespondTo initiated by an event or from the response to a diagnostic request), the client shall present the response both as a serviceToRespondTo-response and as the response to the diagnostic request. The client shall not repeat the request with the exception of NegativeResponseCode busyRepeatRequest (0x21) (see the negative response code definitions in ISO 14229-1).

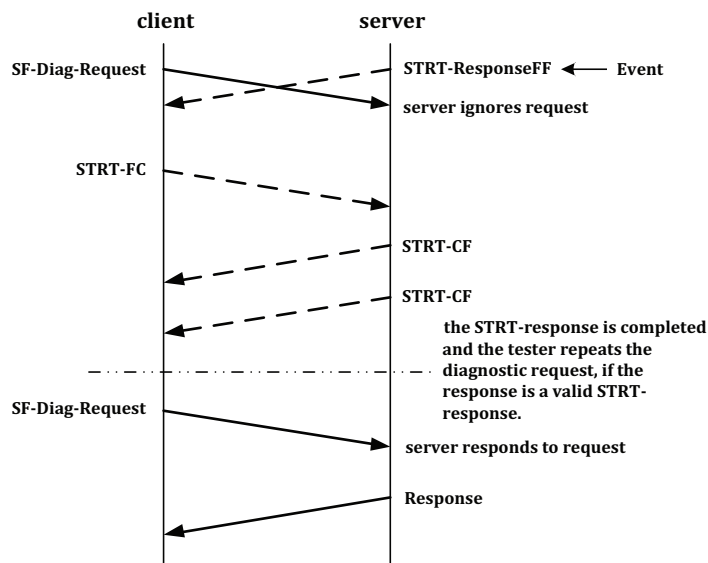


Figure 2 — Concurrent request when the event occurs

- c) When the specified event is detected, the server shall respond immediately with the appropriate serviceToRespondTo-response message. The immediate serviceToRespondTo-response message shall not disrupt any other diagnostic request or response transmission already in progress (i.e. the serviceToRespondTo-response shall be delayed until the current message transmission has been completed, see Figure 3).

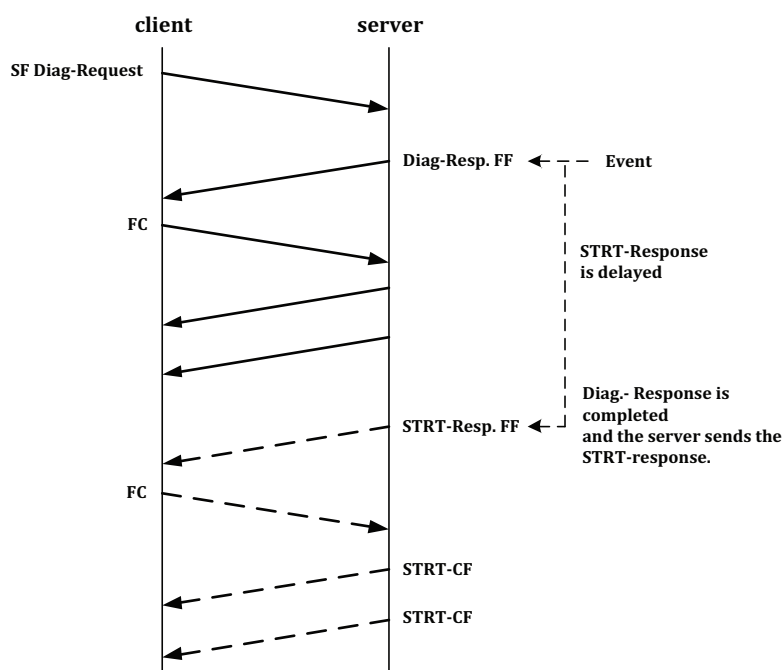


Figure 3 — Event occurrence during a message in progress

## 6.4 ReadDataByPeriodicIdentifier (0x2A) service

### 6.4.1 Periodic transmission response message

The ReadDataByPeriodicIdentifier service allows the client to request the periodic transmission of data record values from the server identified by one or more periodicDataIdentifiers. For service 0x2A implementation on DoCAN, the periodic transmission response message is supported according to ISO 14229-1.

The usage of the periodic transmission response message should consider a single data format being supported for the whole vehicle which may consist of other data links than CAN. E.g. if FlexRay is part of the electrical vehicle architecture in addition to CAN, the total DID data length shall not exceed the length limitations of the CAN protocol to ensure a single data format.

Table 3 describes the periodic transmission response message.

**Table 3 — Periodic transmission — Requirements for response message mapping**

Message type	Client request requirements	Server response requirements	Further server restrictions
UUDT uses a different CAN Identifier for periodic transmission	No restrictions	Only single-frame responses for periodic transmission Multi-frame responses to new (non-periodic-transmission) requests are possible	The request for periodic transmission is processed as a regular diagnostic request and the response is sent via the network layer (as a USDT message with service identifier 0x6A).
			On receiving the N_USData.con that indicates the completion of the transmission of the positive response, the application starts an independent scheduler, which handles the periodic transmission.
			The scheduler in the server processes the periodic transmission as a single CAN frame response message in a bypass (i.e. writes the message directly to the CAN-controller/data link layer driver without using the network-layer).
			There is neither a protocol control information (PCI) nor a service identifier (Sid) included in the response message. Only the periodic identifier and corresponding data are included.

### 6.4.2 Periodic transmission response message handling

#### 6.4.2.1 General

Due to the fact, that the periodic response message neither supports protocol control information nor the service identifier information (transport layer by-pass), the following service primitives need to be taken into account. The service primitives as defined below make use of the individual parameters as specified in ISO 15765-2:

#### 6.4.2.2 N\_UUData.request

The service primitive requests periodic transmission of < MessageData > with < Length > number of bytes from the sender to the receiver peer entities identified by the address information in N\_SA and N\_TA. Each time the N\_UUData.request service is called, the transport/network layer shall signal the completion (or failure) of the message transmission to the service user by means of the issuing of a N\_UUData.confirm service call:

```

N_UUData.request      (
                        N_SA,
                        N_TA,
                        <Length>,
                        <MessageData>
                        )

```

#### 6.4.2.3 N\_UUData.confirm

The N\_UUData.confirm service is issued by the transport/network layer. The service primitive confirms the completion of a N\_UUData.request service identified by the address information in N\_SA and N\_TA. The parameter < N\_Result > provides the status of the service request (referring < N\_Result > values defined in ISO 15765-2).

```

N_UUData.confirm      (
                        N_SA,
                        N_TA,
                        <N_Result>
                        )

```

NOTE Not all result values as defined in ISO 15765-2 apply for the response message transmission type UUDT.

#### 6.4.2.4 N\_UUData.indication

The N\_UUData.indication service is issued by the transport/network layer. The service primitive delivers < MessageData > with < Length > bytes received from a peer protocol entity identified by the address information in N\_SA and N\_TA to the upper layer (see definition in ISO 15765-2).

The parameters < MessageData > and < Length > are only valid when the service primitive is indicated. In case of a reception error no indication will be generated. Therefore a < N\_Result > parameter is not needed.

```

N_UUData.indication    (
                        N_SA,
                        N_TA,
                        <Length>,
                        <MessageData>
                        )

```

### 6.4.3 DoCAN frame format

#### 6.4.3.1 General

Since the periodic transmission response message neither supports PCI information nor service identification information the protocol data unit format (PDU) differs to what is defined in ISO 15765-2. However, the CAN periodic transmission PDU format consists of three fields, as specified in Table 4.

**Table 4 — Periodic transmission — PDU format**

Address information	Length information	Data field
CAN Identifier C_AI	DLC	<MessageData>

#### 6.4.3.2 Address information

The C\_AI is used to identify the communicating peer entities for the periodic message PDUs. The exchange of network layer data are supported by normal addressing. The address information is mapped onto the CAN Identifier see ISO 15765-2.

#### 6.4.3.3 Length Information

The DLC parameter specifies the number of data bytes transmitted in a CAN frame.

#### 6.4.3.4 Data Field

The < MessageData > field contains periodic data information defined with a periodicDataIdentifier (PDID) and its corresponding dataRecords. The maximum size of the < MessageData > field shall be eight bytes.

#### 6.4.4 Periodic transmission message flow

Figure 4 graphically illustrates the periodic response messages handling as specified above. As already stated in 6.4.2, the figure shows that the periodically transmitted response messages do not have any influence on the S3<sub>Server</sub> timer of the server. For this figure it is assumed that a non-defaultSession has been activated prior to the configuration of the periodic scheduler (the ReadDataByPeriodicIdentifier service requires a non-defaultSession in order to be executed).





**Key**

- 1 **Client T\_Data.req:** The diagnostic application of the client starts the transmission of the ReadDataByPeriodicIdentifier (0x2A) request message by issuing a T\_Data.req to its transport/network layer. The transport/network layer transmits the ReadDataByPeriodicIdentifier (0x2A) request message to the server. The request message can either be a segmented or unsegmented message (depends on the number of periodicDataIdentifier contained in the request message). For the example given, it is assumed that the request message fits into a single frame
- 2 **Client T\_Data.con:** The completion of the request message is indicated in the client via T\_Data.con. Now the response timing as described in ISO 14229-2 applies  
  
**Server T\_Data.ind:** The completion of the request message is indicated in the server via the T\_Data.ind. Now the response timing as described in ISO 14229-2 applies. Furthermore, the server stops its S3<sub>Server</sub> timer
- 3 **Server T\_Data.req:** It is assumed that the client requires a response from the server. The server shall transmit the ReadDataByPeriodicIdentifier positive response message to indicate that the request has been processed
- 4 **Server T\_Data.con:** The completion of the transmission of the ReadDataByPeriodicIdentifier response message is indicated in the server via T\_Data.con. Now the server restarts its S3<sub>Server</sub> timer, which keeps the activated non-default session active as long as it does not time out. The transmission of the periodic messages will now be started  
  
**Client T\_Data.ind:** The reception of the response message is indicated in the client
- 5 **Server N\_UUData.req:** The server starts to transmit the periodic response messages (single DoCAN frame messages). Each periodic message which neither includes any PCI information nor service identification uses a different source address than USDTC response messages. Therefore, the server issues a N\_UUData.req each time a periodic message is transmitted independent of any other service currently processed by the server. This means that the transmission of the periodic response messages continues even when the server is in the process of handling another diagnostic service request. The transmission of the periodic response messages has no influence on the S3<sub>Server</sub> timer
- 6 **Server N\_UUData.con:** The completion of the transmission of the periodic response message is indicated in the server  
  
**Client N\_UUData.ind:** The completion of the reception of the periodic response message is indicated in the client
- 7 See (5)
- 8 See (6)
- 9 See (5)
- 10 See (6)
- 11 **Client T\_Data.req:** The diagnostic application of the client starts the transmission of the next request message by issuing a T\_Data.req to its transport/network layer. The transport/network layer transmits the request message to the server. The request message can either be an unsegmented or segmented message. For the example given, it is assumed that the request message is a multi-frame message
- 12 **Server T\_Data\_SOM.ind:** The start of a request message is indicated in the server via T\_Data\_SOM.ind while a periodic scheduler is active. The server does not stop the periodic scheduler for the duration of processing the received request message. This means that the server transmits further periodic messages for the duration of processing the diagnostic service. The client shall be aware of receiving these periodic response messages. Furthermore, any time the server is in the process of handling any diagnostic service it stops its S3<sub>Server</sub> timer
- 13 **Client T\_Data.con:** The completion of the request message is indicated in the client via T\_Data.con. Now the response timing as described in ISO 14229-2 applies

**Server T\_Data.ind:** The completion of the multi-frame request message is indicated in the server via the T\_Data.ind. Now the response timing as described in ISO 14229-2 applies

14 See (5)

15 See (6)

16 **Server T\_Data.req:** It is assumed that the client requires a response from the server. The server shall transmit the positive (or negative) response message via issuing a T\_Data.req to its transport/network layer. In this example it is assumed that the response is a multi-frame message. While the multi-frame response message is transmitted by the transport/network layer, the periodic scheduler continues to transmit the periodic response messages

17 **Client T\_Data\_SOM.ind:** The start of the response message is indicated in the client

18 See (5)

19 See (6)

20 **Client T\_Data.req:** When the S3<sub>Client</sub> timer times out in the client, then the client transmits a functionally addressed TesterPresent (0x3E) request message to restart the S3<sub>Server</sub> timer in the server

21 **Server T\_Data.ind:** The server is in the process of transmitting the multi-frame response message of the previous request. Therefore, the server shall not act on the received TesterPresent (0x3E) request message, because its S3<sub>Server</sub> timer is not yet re-activated

**Client T\_Data.con:** The reception of the TesterPresent (0x3E) request message is indicated in the client

22 See (5)

23 See (6)

24 **Server T\_Data.con:** When the diagnostic service is completely processed, then the server restarts its S3<sub>Server</sub> timer. This means that any diagnostic service, including TesterPresent (0x3E), restarts the S3<sub>Server</sub> timer. A diagnostic service is meant to be in progress any time between the start of the reception of the request message (T\_Data\_SOM.ind or T\_Data.ind receive) and the completion of the transmission of the response message, where a response message is required or the completion of any action that is caused by the request, where no response message is required (point in time reached that would cause the start of the response message). This includes negative response messages including response code 0x78

**Client T\_Data.ind:** The reception of the response message is indicated in the client

25 See (5)

26 See (6)

27 See (5)

28 See (6)

29 **Client T\_Data.req:** When the S3<sub>Client</sub> timer times out in the client (non-defaultSession active), then the client transmits a functionally addressed TesterPresent (0x3E) request message which does not require a response message, each time the S3<sub>Client</sub> timer times out

30 **Client T\_Data.con:** Upon the indication of the completed transmission of the TesterPresent (0x3E) request message via T\_Data.con of its transport/network layer, the client once again starts its S3<sub>Client</sub> timer. This means that the functionally addressed TesterPresent (0x3E) request message is sent on a periodic basis every time S3<sub>Client</sub> times out

**Server T\_Data.ind:** The reception of the TesterPresent (0x3E) request message is indicated in the server. The server shall re-activate the S3<sub>Server</sub> timer

**Figure 4 — Periodic transmission response message handling**

## 7 Application layer requirements

### 7.1 Application layer services

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

### 7.2 Application layer protocol

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

### 7.3 Application layer timing

#### 7.3.1 General

The subclauses specify the message timing parameters and how they apply to the client and the server.

#### 7.3.2 Message timing parameter values

The application layer timing parameter values shall be in accordance with the definitions in ISO 14229-2.

#### 7.3.3 Unsolicited response messages

Unsolicited messages are those transmitted by the server(s) based on either a periodic scheduler (see service ReadDataByPeriodicIdentifier in 6.4) or a configured trigger, such as a change of a DTC status or a dataIdentifier value change (see service ResponseOnEvent in 6.3).

Any unsolicited transmitted response message shall not reset the  $S3_{Server}$  timer in the server. This avoids a diagnostic session keep-alive latch-up effect in the server for cases where a periodic message transmission is active or a timer-triggered event is configured in the server where the time interval between the events is smaller than  $S3_{Server}$ . The  $S3_{Server}$  timer shall only be reset if the transmitted response message is the direct result of processing a request message and transmitting the final response message (such as the initial positive response that indicates that a request to schedule one or more periodicDataIdentifiers is performed successfully).

## 8 Presentation layer requirements

The presentation layer requirements are in the responsibility of the vehicle manufacturer.

## 9 Session layer requirements

The session layer requirements are specified in ISO 14229-2.

## 10 Transport/Network layer interface adaptation

### 10.1 General information

This part of ISO 14229 makes use of the network layer services defined in ISO 14229-2 for the transmission and reception of diagnostic messages. This subclause defines the mapping of the data link independent transport/network layer protocol data units (T\_PDU) onto the CAN data link specific network layer protocol data units (N\_PDU).

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

## 10.2 DoCAN Transport/Network layer interface adaptation

### 10.2.1 Mapping of data link independent service primitives onto CAN data link dependent service primitives

Table 5 specifies the mapping interface between ISO 15765-2, DoCAN transport protocol and network layer services and the session layer services defined in ISO 14229-2 for the transmission and reception of diagnostic messages.

**Table 5 — Mapping of T\_PDU service primitives onto N\_PDU service primitives**

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

### 10.2.2 Mapping of T\_PDU onto N\_PDU for message transmission

The parameters of the application layer protocol data unit defined to request the transmission of a diagnostic service request/response are mapped in accordance with Table 6 onto the parameters of the transport/network layer protocol data unit for the transmission of a message in the client/server.

**Table 6 — Mapping of T\_PDU parameter onto N\_PDU parameter**

T_PDU parameter (data link independent according to ISO 14229-2)	N_PDU parameter (CAN data link dependent according to ISO 15765-2)
T_Mtype	N_Mtype
T_SA	N_SA
T_TA	N_TA
T_TAtype	N_TAtype
T_AE  Mtype = diagnostics, then the address information shall consist of the parameters SA, TA, and TAtype.  If Mtype = remote diagnostics, then the address information shall consist of the parameters SA, TA, TAtype, and AE.	N_AE
T_Data [ ]	<MessageData>
T_Length	<Length>
T_Result	<N_Result>

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

The address mapping between the network layer and the OSI higher layers is not necessarily an exact copy of the address values as encoded on the data link layer and therefore depends on the implementation concept.

The mapping is defined in ISO 15765-2. Values for emissions-related OBD are defined in ISO 15765-4.

## 11 Data link layer diagnostic implementation requirements

### 11.1 Usage of legislated 11-bit OBD CAN Identifiers for enhanced diagnostics

The 11-bit CAN Identifiers for legislated OBD can also be used for enhanced diagnostics (e.g. the functional request CAN Identifier can be used for the functionally addressed TesterPresent (0x3E) request message to keep a non-defaultSession active).

If the 11-bit CAN identifiers are used for enhanced diagnostics as specified in ISO 15765-4, then the following requirements apply:

- a) the network layer timing parameters according ISO 15765-4 shall also apply for enhanced diagnostics;
- b) the DLC (CAN data length code) shall be set to eight and the CAN frame shall include eight bytes (unused bytes shall be padded).

### 11.2 Usage of legislated 29-bit OBD CAN Identifiers for enhanced diagnostics

The 29-bit CAN Identifiers for emissions-related legislated OBD servers comply with the normal fixed addressing format specified in ISO 15765-2 and can also be used for enhanced diagnostics.

If the 29-bit CAN identifiers as specified in ISO 15765-4 are re-used for enhanced diagnostics, then the following requirements apply:

- a) the network layer timing parameters as specified in ISO 15765-4 shall also apply for enhanced diagnostics;
- b) the DLC shall be set to eight and the CAN frame shall include eight bytes (unused bytes shall be padded).

NOTE The CAN Identifier values given in the ISO 15765-2 use the default value for the priority information.

## Bibliography

- [1] ISO/IEC 7498-1, *Information technology — Open Systems Interconnection — Basic Reference Model — The Basic Mode*
- [2] ISO/IEC 10731:1994, *Information technology — Open Systems Interconnection — Basic Reference Model — Conventions for the definition of OSI services*
- [3] ISO 11898-1, *Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signalling*
- [4] ISO 11898-2, *Road vehicles — Controller area network (CAN) — Part 2: High-speed medium access unit*
- [5] ISO 11898-3, *Road vehicles — Controller area network (CAN) — Part 3: Low-speed, fault-tolerant, medium dependent interface*
- [6] ISO 11898-5, *Road vehicles — Controller area network (CAN) — Part 5: High-speed medium access unit with low-power mode*
- [7] ISO 13400 (all parts), *Road vehicles — Communication between test equipment and vehicles over Internet Protocol*
- [8] SAE J1930-DA, *Digital Annex of Electrical/Electronic Systems Diagnostic Terms, Definitions, Abbreviations, and Acronyms Web Tool Spreadsheet*
- [9] SAE J1939, *Companion Spreadsheet (CS1939)*
- [10] SAE J1939-73:2010, *Recommended Practice for a Serial Control and Communication Vehicle Network — Application layer — Diagnostics*
- [11] SAE J1979-DA, *Digital Annex of E/E Diagnostic Test Modes*
- [12] SAE J2012-DA, *Digital Annex of Diagnostic Trouble Code Definitions and Failure Type Byte Definitions*



