

---

---

**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)*

STANDARDSISO.COM : Click to view the full PDF of ISO 14229-3:2022



STANDARDSISO.COM : Click to view the full PDF of ISO 14229-3:2022



**COPYRIGHT PROTECTED DOCUMENT**

© ISO 2022

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
CP 401 • Ch. de Blandonnet 8  
CH-1214 Vernier, Geneva  
Phone: +41 22 749 01 11  
Email: [copyright@iso.org](mailto:copyright@iso.org)  
Website: [www.iso.org](http://www.iso.org)

Published in Switzerland

# Contents

	Page
Foreword.....	v
Introduction.....	vi
<b>1 Scope.....</b>	<b>1</b>
<b>2 Normative references.....</b>	<b>1</b>
<b>3 Terms and definitions.....</b>	<b>1</b>
<b>4 Symbols and abbreviated terms.....</b>	<b>1</b>
4.1 Symbols.....	1
4.2 Abbreviated terms.....	2
<b>5 Conventions.....</b>	<b>2</b>
<b>6 Service primitive interface definition.....</b>	<b>3</b>
<b>7 Technical requirements overview.....</b>	<b>3</b>
<b>8 Application layer.....</b>	<b>4</b>
8.1 ISO 14229-1 service primitive parameters.....	4
8.2 A_Data.req, A_Data.ind, and A_Data.conf service interface.....	4
8.3 UDSONCAN services overview.....	4
8.4 A_PDU definition.....	5
8.5 ReadDataByPeriodicIdentifier service UDSONCAN implementation requirements.....	6
8.5.1 UUDT periodic transmission response message handling.....	6
8.5.2 Service interface – UUDT.....	6
8.5.3 UUDT service primitive parameters.....	8
8.5.4 UUDT message format.....	9
8.5.5 Periodic transmission message flow.....	10
8.6 Timing parameter definition.....	13
8.6.1 Request and response message timing parameter values.....	13
8.6.2 Unsolicited response messages.....	13
<b>9 Presentation layer.....</b>	<b>13</b>
<b>10 Session layer.....</b>	<b>13</b>
10.1 Service primitive parameter definition.....	13
10.2 S_Data.req, S_Data.ind, and S_Data.conf service interface.....	13
<b>11 Transport layer.....</b>	<b>14</b>
11.1 USDT service primitive parameters.....	14
11.2 T_Data.req, T_Data.ind, and T_Data.conf service interface.....	14
11.3 Transport protocol.....	14
11.4 T_PDU definition.....	14
11.5 DoCAN transport and network layer interface adaptation.....	14
11.5.1 Mapping of data link independent service primitives onto CAN data link- dependent service primitives.....	14
11.5.2 Mapping of T_PDU onto N_PDU.....	15
<b>12 Network layer.....</b>	<b>15</b>
12.1 Service primitive parameter definition.....	15
12.2 N_Data.req, N_Data.ind, and N_Data.conf service interface.....	15
12.3 Network layer services.....	16
12.4 N_PDU definition.....	16
12.5 N_TAtype service primitive parameter.....	16
12.6 Same N_TAtype request and associated response message format.....	16
<b>13 Data link layer.....</b>	<b>17</b>
13.1 Service primitive parameter definition.....	17
13.2 L_Data.req, L_Data.ind, and L_Data.conf service interface.....	17
13.3 Usage of ISO 15765-4-defined 11-bit CAN identifiers for enhanced diagnostics.....	17

13.4	Usage of ISO 15765-4-defined 29-bit CAN identifiers for enhanced diagnostics.....	18
<b>14</b>	<b>Physical layer.....</b>	<b>18</b>
	<b>Bibliography.....</b>	<b>19</b>

STANDARDSISO.COM : Click to view the full PDF of ISO 14229-3:2022

## Foreword

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

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

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see [www.iso.org/patents](http://www.iso.org/patents)).

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

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see [www.iso.org/iso/foreword.html](http://www.iso.org/iso/foreword.html).

This document was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 31, *Data communication*.

This second edition cancels and replaces the first edition (ISO 14229-3:2012), which has been technically revised.

The main changes are as follows:

- restructuration of the document;
- introduction of requirement numbers, names and definitions;
- technical content improvements based on implementation feedback from the automotive industry.

A list of all parts in the ISO 14229 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at [www.iso.org/members.html](http://www.iso.org/members.html).

## Introduction

The ISO 14229 series has been established in order to define common requirements for diagnostic systems, whatever the serial data link is.

To achieve this, the ISO 14229 series is based on the Open Systems Interconnection (OSI) Basic Reference Model in accordance with ISO/IEC 7498-1<sup>[1]</sup> and ISO/IEC 10731<sup>[2]</sup>, which structures communication systems into seven layers. When mapped on this model, the services used by a diagnostic tester (client) and an Electronic Control Unit (ECU, server) are structured into the following layers:

- application layer (layer 7) specified in ISO 14229-1 and ISO 14229-3 to ISO 14229-8;
- presentation layer (layer 6) specified in ISO 14229-1 and ISO 14229-3 to ISO 14229-8;
- session layer services (layer 5) specified in ISO 14229-2 and ISO 14229-3 to ISO 14229-8.

Figure 1 illustrates the UDSONCAN document and related documents according to the OSI model.

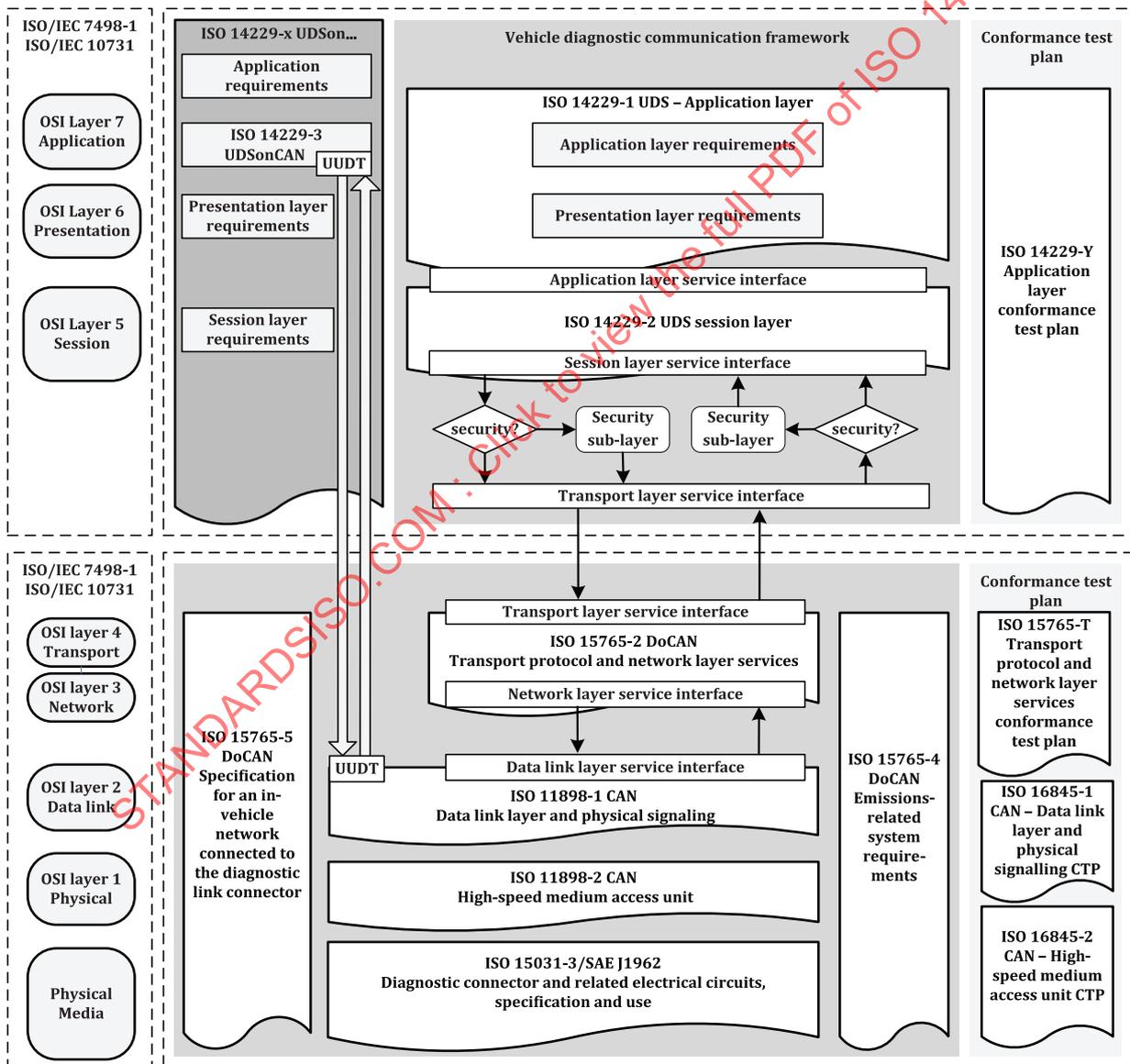


Figure 1 — ISO 14229-3 document reference according to OSI model

# Road vehicles — Unified diagnostic services (UDS) —

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

### 1 Scope

This document specifies an application profile for the implementation of unified diagnostic services (UDS) on controller area network (CAN) in road vehicles.

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

This document specifies

- 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 are referred to in the text in such a way that some or all of their content constitutes requirements 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 11898-1, *Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signalling*

ISO 14229-1, *Road vehicles — Unified diagnostic services (UDS) — Part 1: Application layer*

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

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

ISO 15765-5, *Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) — Part 5: Specification for an in-vehicle network connected to the diagnostic link connector*

### 3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 14229-1, ISO 14229-2, ISO 15765-2, and ISO 15765-5 apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

### 4 Symbols and abbreviated terms

#### 4.1 Symbols

—	empty table cell or feature undefined
$t$	time
$t_{P\_Client}$	client application layer timer
$t_{P2\_Server}$	server application layer timer
$t_{P2\_CAN\_Client}$	client application layer timeout value for CAN
$t_{P2\_CAN\_Server}$	server application layer timeout value for CAN
$t_{S3\_Client}$	client session layer timer
$t_{S3\_Server}$	server session layer timer
$t_{S3\_Server\_Reload}$	server session layer timeout-reload value

## 4.2 Abbreviated terms

CAN	Controller Area Network
CBFF	classical base frame format
CEFF	classical extended frame format
DA	destination address
DLC	data length code
FBFF	FD base frame format
FEFF	FD extended frame format
IVN	in-vehicle network
PCI	protocol control information
SA	source address
SId	service identifier
SOM	start of message
STRT	serviceToRespondTo
TA	target address
UDS	unified diagnostic services
USDT	unacknowledged segmented data transfer
UUDT	unacknowledged unsegmented data transfer

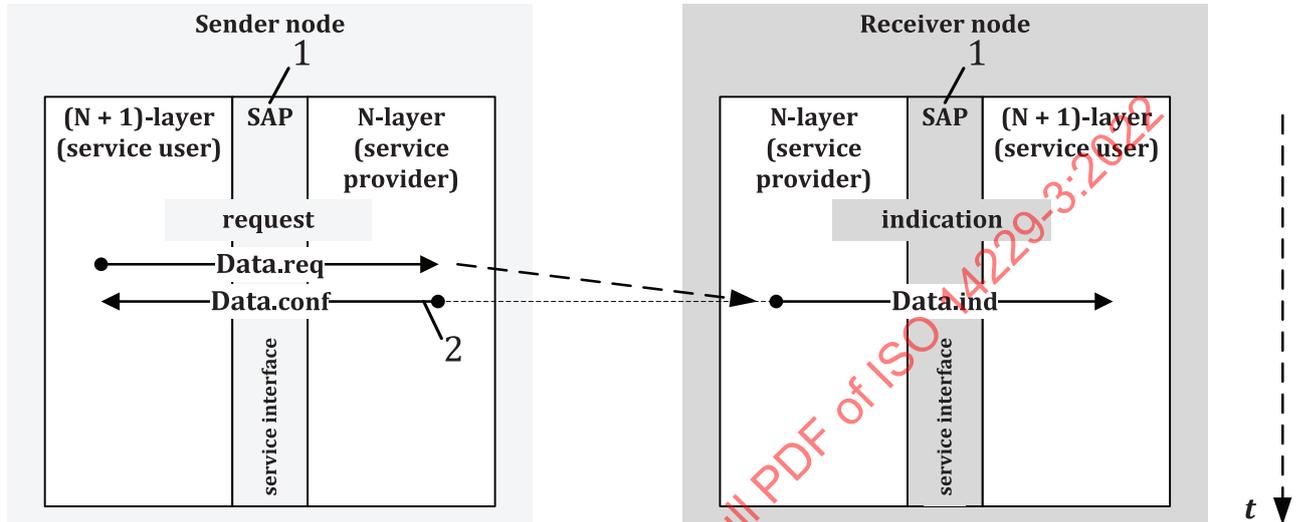
## 5 Conventions

This document is based on OSI service conventions as specified in ISO/IEC 10731<sup>[2]</sup>.

## 6 Service primitive interface definition

The service interface defines the service primitive from the application layer to the session layer.

Figure 2 shows the Data.req (request), Data.ind (indication), and Data.conf (confirmation) service interface.



### Key

- 1 service access point between application and application layer
- 2 read back from N-layer service provider
- t time

Figure 2 — Data.req, Data.ind, and Data.conf service interface

## 7 Technical requirements overview

Table 1 provides an overview about technical requirements and associated requirement numbers.

Table 1 — Technical requirements overview

OSI#.REQ#	Technical requirement title
7	<b>Application layer</b>
7.1	ISO 14229-1 service primitive parameters
7.2	A_Data.req, A_Data.ind, and A_Data.conf service interface
7.3	UDSonCAN specific requirements
7.4	No UDSonCAN-specific requirements
7.5	UUDT periodic transmission response message handling
7.6	UUDT periodic transmission response message server restrictions
7.7	UUDT OSI-layer support
7.8	Service primitive - A_UUDData.req
7.9	Service primitive - A_UUDData.ind
7.10	Service primitive - A_UUDData.conf
7.11	General UUDT service primitive parameters
7.12	Specific UUDT service primitive parameters

**Table 1** (continued)

OSI#.REQ#	Technical requirement title
7.13	UUDT message format
7.14	UUDT A_PDU size
7.15	Request and response message timing parameter values
7.16	Unsolicited response messages
<b>6</b>	<b>Presentation layer</b>
---	No requirement statement in this document
<b>5</b>	<b>Session layer</b>
5.1	Service primitive parameter definition
5.2	S_Data.req, S_Data.ind, and S_Data.conf service interface
<b>4</b>	<b>Transport layer</b>
4.1	USDT service primitive parameters
4.2	T_Data.req, T_Data.ind, and T_Data.conf service interface
<b>3</b>	<b>Network layer</b>
3.1	Service primitive parameter definition
3.2	N_Data.req, N_Data.ind, and N_Data.conf service interface
3.4	N_TAtype service primitive parameter
3.5	Same N_TAtype request and associated response message format
<b>2</b>	<b>Data link layer</b>
2.1	Service primitive parameter definition
2.2	L_Data.req, L_Data.ind, and L_Data.conf service interface
<b>1</b>	<b>Physical layer</b>
—	No requirement statement in this document.

## 8 Application layer

### 8.1 ISO 14229-1 service primitive parameters

<b>REQ</b>	<b>7.1 UDSONCAN - ISO 14229-1 service primitive parameters</b>
The service primitive parameter shall be implemented as specified in ISO 14229-1.	

### 8.2 A\_Data.req, A\_Data.ind, and A\_Data.conf service interface

This document is part of the ISO 14229 series and therefore, the service interface implementation follows the ISO 14229-1 specification.

<b>REQ</b>	<b>7.2 UDSONCAN - A_Data.req, A_Data.ind, and A_Data.conf service interface</b>
The A_Data.req, A_Data.ind, and A_Data.conf service interface shall be implemented as specified in ISO 14229-1.	

### 8.3 UDSONCAN services overview

The purpose of [Table 2](#) is to reference ISO 14229-1 and ISO 14229-2 services as they are applicable for an implementation in this document. [Table 2](#) contains the UDSONCAN applicable diagnostic services. Certain UDSONCAN applications can restrict the number of useable services and can categorize them in application areas/diagnostic sessions (default session, programming session, etc.).

<b>REQ</b>	<b>7.3 UDSONCAN specific requirements</b>
------------	-------------------------------------------

Services that are marked “UDSonCAN-specific requirements” shall be implemented as specified in the referenced subclause number in accordance with [Table 2](#) “Reference” column.

<b>REQ</b>	<b>7.4 No UDSonCAN-specific requirements</b>
Services specified in <a href="#">Table 2</a> that are marked “No UDSonCAN-specific requirements” shall be implemented as specified in ISO 14229-1 and ISO 14229-2 with no additional restrictions.	

**Table 2 — Overview of applicable ISO 14229-1-defined services**

Functional unit name	Diagnostic service name	Comment	Reference
<b>Diagnostic and communication management</b>	DiagnosticSessionControl	No UDSonCAN-specific requirements	—
	ECUReset	No UDSonCAN-specific requirements	—
	SecurityAccess	No UDSonCAN-specific requirements	—
	CommunicationControl	No UDSonCAN-specific requirements	—
	TesterPresent	No UDSonCAN-specific requirements	—
	AccessTimingParameters	Not supported	—
	Authentication	No UDSonCAN-specific requirements	—
	SecuredDataTransmission	No UDSonCAN-specific requirements	—
	ControlDTCSetting	No UDSonCAN-specific requirements	—
	ResponseOnEvent	No UDSonCAN-specific requirements	—
LinkControl	No UDSonCAN-specific requirements	—	
<b>Data transmission</b>	ReadDataByIdentifier	No UDSonCAN-specific requirements	—
	ReadMemoryByAddress	No UDSonCAN-specific requirements	—
	ReadScalingDataByIdentifier	No UDSonCAN-specific requirements	—
	ReadDataByPeriodicIdentifier	UDSonCAN-specific requirements	see <a href="#">8.5</a>
	DynamicallyDefineDataIdentifier	No UDSonCAN-specific requirements	—
	WriteDataByIdentifier	No UDSonCAN-specific requirements	—
<b>Stored data transmission</b>	WriteMemoryByAddress	No UDSonCAN-specific requirements	—
	ReadDTCInformation	No UDSonCAN-specific requirements	—
<b>Input/output control</b>	ClearDiagnosticInformation	No UDSonCAN-specific requirements	—
	InputOutputControlByIdentifier	No UDSonCAN-specific requirements	—
<b>Remote activation of routine</b>	RoutineControl	No UDSonCAN-specific requirements	—
<b>Upload/download</b>	RequestDownload	No UDSonCAN-specific requirements	—
	RequestUpload	No UDSonCAN-specific requirements	—
	TransferData	No UDSonCAN-specific requirements	—
	RequestTransferExit	No UDSonCAN-specific requirements	—
	RequestFileTransfer	No UDSonCAN-specific requirements	—

### 8.4 A\_PDU definition

[Figure 3](#) shows the A\_PDU.

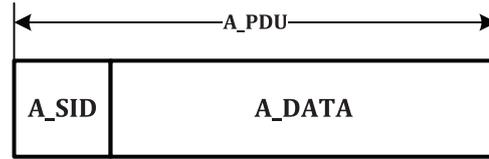


Figure 3 — A\_PDU definition

## 8.5 ReadDataByPeriodicIdentifier service UDSONCAN implementation requirements

### 8.5.1 UUDT periodic transmission response message handling

The periodic transmission response message is based on a message size which is supported by the in-vehicle network (IVN). The IVN can consist of gateways and other connected data links than CAN.

<b>REQ</b>	<b>7.5 UDSONCAN – ReadDataByPeriodicIdentifier – UUDT periodic transmission response message handling</b>
The A_PDU length referenced by a periodicDataIdentifier (pDID) shall not exceed the length limitation of a non-segmented message.	

<b>REQ</b>	<b>7.6 UDSONCAN – ReadDataByPeriodicIdentifier – UUDT periodic transmission response message server restrictions</b>
Client receiving periodic response messages shall support the message mapping as specified in <a href="#">Table 3</a> . Server transmitting periodic response messages shall support the message mapping as specified in <a href="#">Table 3</a> .	

Table 3 — Periodic transmission response message mapping

Message type	Client request requirements	Server response requirements	Further server restrictions
A UUDT message shall use a different CAN identifier for periodic transmission.	No restrictions	1. Only single-frame responses for periodic transmission 2. 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 6A <sub>16</sub> ).
			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.

### 8.5.2 Service interface – UUDT

#### 8.5.2.1 General

The unconfirmed unsegmented data transfer (UUDT) is a response message type. UUDT messages are communicated between the application layer and the data link layer. UUDT periodic response

messages neither support the service identifier information (application layer) nor the protocol control information (transport layer).

<b>REQ</b>	<b>7.7 UDSONCAN - ReadDataByPeriodicIdentifier - UUDT OSI-layer support</b>
UUDT messages shall bypass the session layer, transport layer, and the network layer and implemented in the data link layer.	

### 8.5.2.2 Service primitive - A\_UUData.req

The A\_UUData.req service primitive is issued by the sender node.

<b>REQ</b>	<b>7.8 UDSONCAN - UUDT service primitive - A_UUData.req</b>
The service primitives shall request periodic transmission of A_Data with A_Length number of bytes from the sender to the receiver peer entities identified by the message type A_Mtype and address information in A_UUTAtype, A_SA and A_TA.	

```
A_UUData.req
(
    A_Mtype,
    A_SA,
    A_TA,
    A_UUTAtype,
    A_Data[Data#1, Data#2, ..., Data#n ],
    A_Length,
)
```

### 8.5.2.3 Service primitive - A\_UUData.ind

The A\_UUData.ind service primitive is received by the receiver node.

<b>REQ</b>	<b>7.9 UDSONCAN - UUDT service primitive - A_UUData.ind</b>
The service primitives shall deliver A_Data with A_Length bytes received from a peer protocol entity identified by the message type A_Mtype and address information in A_UUTAtype, A_SA and A_TA.	

The parameters A\_Data and A\_Length are only valid when the service primitive is indicated. In case of a reception error no indication is generated.

```
A_UUData.ind
(
    A_Mtype,
    A_SA,
    A_TA,
    A_UUTAtype,
    A_Data[Data#1, Data#2, ..., Data#n ],
    A_Length,
)
```

### 8.5.2.4 Service primitive - A\_UUData.conf

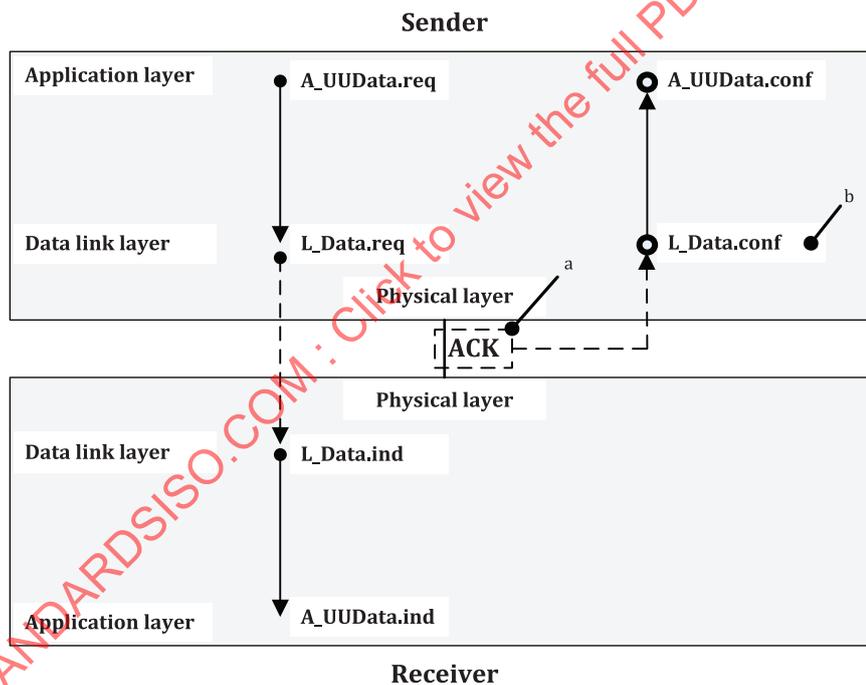
The `A_UUData.conf` service is received by the sender node.

<b>REQ</b>	<b>7.10 UDSonCAN – UUDT service primitive – A_UUData.conf</b>
The service primitives shall confirm the completion of a <code>A_UUData.req</code> service identified by the message type <code>A_Mtype</code> and address information in <code>A_UUTAtype</code> , <code>A_SA</code> , and <code>A_TA</code> . The parameter <code>A_Result</code> provides the status of the service request.	

```

A_UUData.conf (
    A_Mtype,
    A_SA,
    A_TA,
    A_UUTAtype,
    A_Result
    A_Length,
)
    
```

Figure 4 shows the UUDT service primitive interface. The sender node implementation of `L_Data.conf` and `A_UUData.conf` is not in the scope of this document.



- a CAN ACK (acknowledge) bit is set by a receiver node and indicated to the sender.
- b The data link layer of the sender node initiates a `L_Data.conf` service primitive to indicate to the application layer.

Figure 4 — UUDT service primitive interface

### 8.5.3 UUDT service primitive parameters

#### 8.5.3.1 Service primitive data types

UUDT messages are exchanged directly between the application layer and the data link layer.

The service primitive data types derive from ISO 14229-2.

<b>REQ</b>	<b>7.11 UDSonCAN – ReadDataByPeriodicIdentifier – General UUDT service primitive parameters</b>
The data type definitions and the parameters <i>Mtype</i> , <i>TA</i> , <i>SA</i> , <i>Length</i> , <i>Data</i> , and <i>Result</i> shall be implemented as specified in ISO 14229-2.	

### 8.5.3.2 UUDT specific parameters

The parameter *A\_UUDTtype* is UUDT-specific and is a configuration attribute to the *A\_TA* parameter. It is used to encode the UUDT communication model used by the communicating peer entities.

<b>REQ</b>	<b>7.12 UDSonCAN – ReadDataByPeriodicIdentifier – Specific UUDT service primitive parameters</b>
The <i>A_UUDTtype</i> parameter shall be of data type <i>Enum</i> (see ISO 14229-2) and is used to identify the UUDT target address type.	
Range:	
<i>A_UUDTtype</i> #1: physical addressing, UUDT classical base frame format (CBFF), 11-bit CAN identifier, <i>A_UUDTtype</i> #2: physical addressing, UUDT FD base frame format (FBFF), 11-bit CAN identifier, <i>A_UUDTtype</i> #3: physical addressing, UUDT classical extended frame format (CEFF), 29-bit CAN identifier, <i>A_UUDTtype</i> #4: physical addressing, UUDT FD extended frame format (FEFF), 29-bit CAN identifier]	

### 8.5.4 UUDT message format

#### 8.5.4.1 General

A UUDT *A\_PDU* consists of three fields.

<b>REQ</b>	<b>7.13 UDSonCAN – ReadDataByPeriodicIdentifier – UUDT message format</b>
A UUDT message shall be implemented as specified in <a href="#">Table 4</a> .	

**Table 4 — UUDT message (*A\_PDU*) format**

UUDT message identifier	Data field
periodicDataIdentifier	dataRecord

#### 8.5.4.2 UUDT address information

The *A\_AI* (*A\_TA*, *A\_SA*) is provided by the service primitive interface and is used to identify the communicating peer entities for the periodic message (*A\_PDU*).

#### 8.5.4.3 UUDT length information

The length information of the *A\_PDU* is provided by the service primitive interface and is used to identify the length of the UUDT message.

<b>REQ</b>	<b>7.14 UDSonCAN – ReadDataByPeriodicIdentifier – UUDT <i>A_PDU</i> size</b>
The maximum size of the <i>A_PDU</i> data field of the UUDT message shall be in accordance with the parameter <i>TX_DL</i> as specified in ISO 15765-2.	

#### 8.5.4.4 UUDT message identifier

The UUDT message identifier contains the periodicDataIdentifier (pDID).

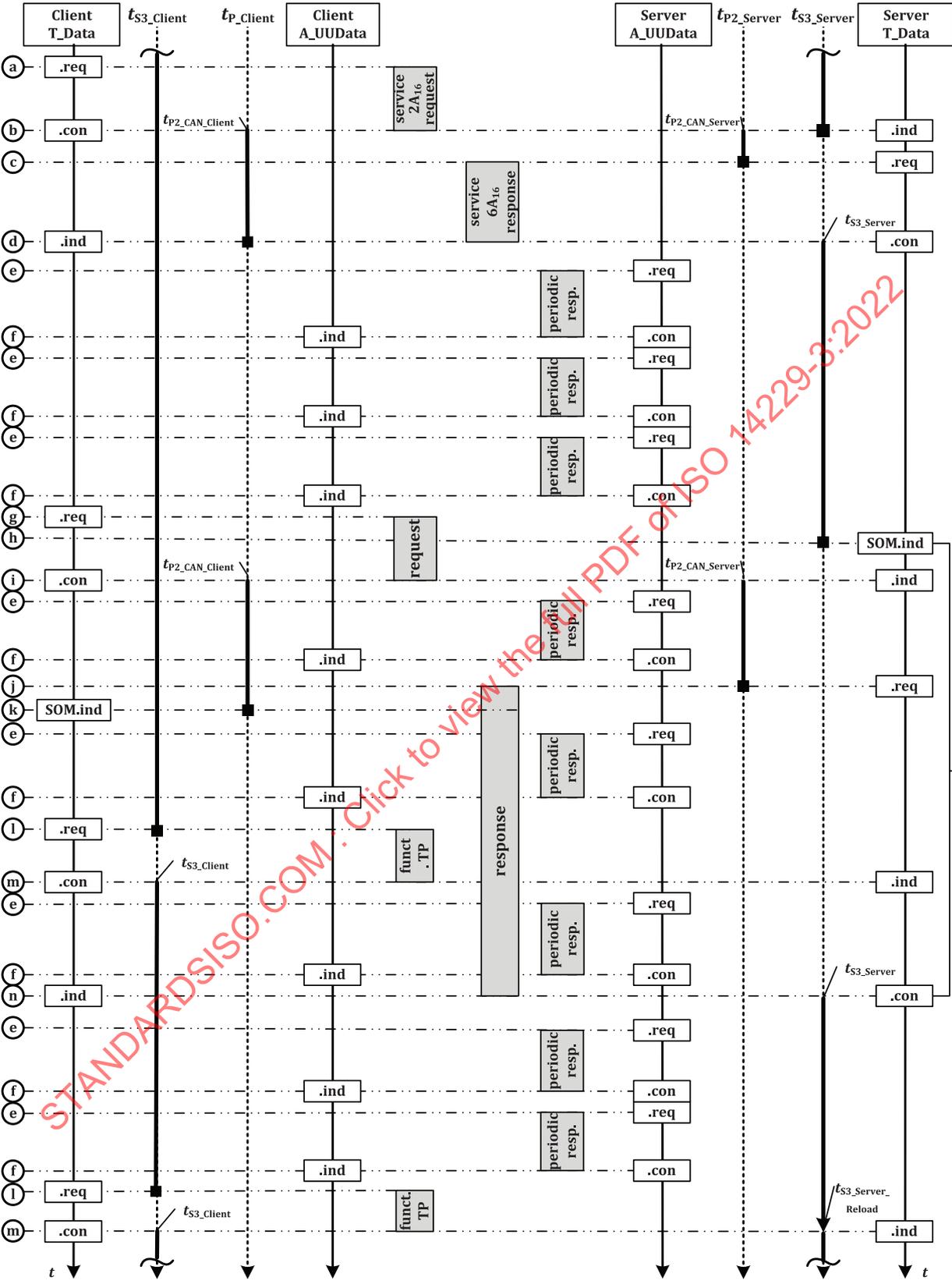
#### 8.5.4.5 UUDT data field information

The data field contains the dataRecord(s) which corresponds to the periodicDataIdentifier (pDID).

#### 8.5.5 Periodic transmission message flow

[Figure 5](#) specifies the periodic response messages handling and shows that the periodically transmitted response messages do not have any influence on the  $t_{S3\_Server}$  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).

STANDARDSISO.COM : Click to view the full PDF of ISO 14229-3:2022



- a Client `T_Data.req`: the diagnostic application of the client starts the transmission of the `ReadDataByPeriodicIdentifier` ( $2A_{16}$ ) request message by issuing a `T_Data.req` to its transport/network layer. The application layer transmits the `ReadDataByPeriodicIdentifier` ( $2A_{16}$ ) request message to the server. The request message can either be a segmented or unsegmented message (depends on the value of the `periodicDataIdentifier` contained in the request message). For the example given, it is assumed that the request message is less or equal eight byte.
- b Client `T_Data.con`: the completion of the request message is indicated in the client via `T_Data.con`. Now the response timing as specified 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 specified in ISO 14229-2 applies. Furthermore, the server stops its `tS3_Server` timer.
- c Server `T_Data.req`: it is assumed that the client requires a response from the server. The server transmits the `ReadDataByPeriodicIdentifier` positive response message to indicate that the request has been processed.
- d 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 `tS3_Server` timer, which keeps the activated non-default session active as long as it does not time out. The transmission of the periodic messages is started.
- Client `T_Data.ind`: the reception of the response message is indicated in the client.
- e Server `A_UUData.req`: the server starts to transmit the periodic response messages. Each periodic message, which neither includes any PCI information nor service identification, uses a different source address than USDT response messages. Therefore, the server issues a `A_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 `tS3_Server` timer.
- f Server `A_UUData.con`: the completion of the transmission of the periodic response message is indicated in the server.
- Client `A_UUData.ind`: the completion of the reception of the periodic response message is indicated in the client.
- g Client `T_Data.req`: the diagnostic application of the client starts the transmission of the next request message by issuing a `T_Data.req`. 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.
- h 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 is aware of receiving these periodic response messages. Furthermore, any time the server is in the process of handling any diagnostic service it stops its `tS3_Server` timer.
- i 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.
- j Server `T_Data.req`: it is assumed that the client requires a response from the server. The server transmits 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.
- k Client `T_Data_SOM.ind`: the start of the response message is indicated in the client.
- l Client `T_Data.req`: when the `tS3_Client` timer times out in the client, then the client transmits a functionally addressed `TesterPresent` ( $3E_{16}$ ) request message to restart the `tS3_Server` timer in the server.
- m Client `T_Data.con`: the reception of the `TesterPresent` ( $3E_{16}$ ) request message is indicated in the client.
- Server `T_Data.ind`: the server is in the process of transmitting the multi-frame response message of the previous request. Therefore, the server does not act on the received `TesterPresent` ( $3E_{16}$ ) request message, because its `tS3_Server` timer is not yet re-activated.
- n Client `T_Data.ind`: the reception of the response message is indicated in the client.

Server `T_Data.con`: when the diagnostic service is completely processed, then the server restarts its  $t_{S3\_Server}$  timer. This means that any diagnostic service, including `TesterPresent (3E16)`, restarts the  $t_{S3\_Server}$  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 `7816`.

- ° Any `TesterPresent` that is received during a disabled  $t_{S3\_Server}$  timer is ignored by the server.

**Figure 5 — Periodic transmission response message handling**

## 8.6 Timing parameter definition

### 8.6.1 Request and response message timing parameter values

The request and response message timing parameters belong to the application layer.

<b>REQ</b>	<b>7.15 UDSonCAN – Request and response message timing parameter values</b>
The request and response message timing parameter values shall be implemented as specified in ISO 14229-2.	

### 8.6.2 Unsolicited response messages

Unsolicited messages are those transmitted by the server(s) based on either a periodic scheduler (see service `ReadDataByPeriodicIdentifier` in 8.5) or a configured trigger, such as a change of a DTC status or a `dataIdentifier` value change.

<b>REQ</b>	<b>7.16 UDSonCAN – Unsolicited response messages</b>
Any unsolicited transmitted response message shall not reset the $t_{S3\_Server}$ 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  $t_{S3\_Server}$ . The  $t_{S3\_Server}$  timer is only reset if the transmitted response message is the 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).

## 9 Presentation layer

The presentation layer specification is not in the scope of this document.

## 10 Session layer

### 10.1 Service primitive parameter definition

<b>REQ</b>	<b>5.1 UDSonCAN – Service primitive parameter definition</b>
The service primitive parameters shall be implemented as specified in ISO 14229-2.	

### 10.2 `S_Data.req`, `S_Data.ind`, and `S_Data.conf` service interface

This document is part of the ISO 14229 series and therefore the service interface implementation follows the ISO 14229-2 specification.

<b>REQ</b>	<b>5.2 UDSONCAN – S_Data.req, S_Data.ind, and S_Data.conf service interface</b>
The S_Data.req, S_Data.ind, and S_Data.conf service interface shall be implemented as specified in ISO 14229-2.	

## 11 Transport layer

### 11.1 USDT service primitive parameters

<b>REQ</b>	<b>4.1 UDSONCAN – USDT service primitive parameters</b>
The service primitive parameters shall be implemented as specified in ISO 15765-5.	

### 11.2 T\_Data.req, T\_Data.ind, and T\_Data.conf service interface

This document supports the service interface definition as specified in ISO 15765-5.

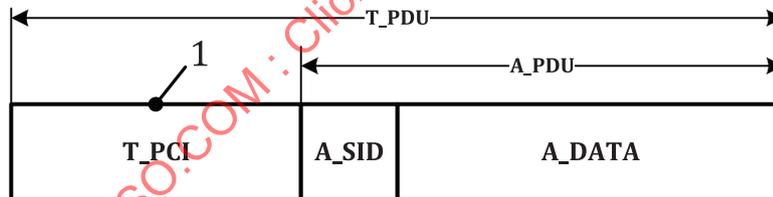
<b>REQ</b>	<b>4.2 UDSONCAN – T_Data.req, T_Data.ind, and T_Data.conf service interface</b>
The T_Data.req, T_Data.ind, and T_Data.conf service interface implementation shall be implemented as specified in ISO 15765-5.	

### 11.3 Transport protocol

<b>REQ</b>	<b>4.3 UDSONCAN – Transport protocol</b>
The transport protocol shall be implemented as specified in ISO 15765-2.	

### 11.4 T\_PDU definition

Figure 6 shows the T\_PDU.



**Key**

1 transport layer protocol control information

Figure 6 — T\_PDU definition

### 11.5 DoCAN transport and network layer interface adaptation

#### 11.5.1 Mapping of data link independent service primitives onto CAN data link-dependent service primitives

<b>REQ</b>	<b>4.4 UDSONCAN – Mapping of data link independent service primitives onto CAN data link-dependent service primitives</b>
The parameter mapping interface shall be implemented as specified in ISO 15765-2, DoCAN transport protocol and network layer services, and the session layer services as specified in ISO 14229-2 for the transmission and reception of diagnostic messages. The parameter mapping interface shall be implemented as specified in Table 5.	