INTERNATIONAL STANDARD

ISO 22900-3

First edition
2009-06-15

Road vehicles — Modular vehicle communication interface (MVCI) —

Part 3: Diagnostic server application programming interface (D-Server API)

Véhicules routiers — Interface de communication modulaire du véhicule (MVCI) —

Partie 3: Interface pour la programmation des applications du serveur de diagnostic (D-Server API)

Reference number
ISO 22900-3:2009(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 22900-3 was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 3, *Electrical and electronic equipment*.

ISO 22900 consists of the following parts, under the general title *Road vehicles — Modular vehicle communication interface (MVCI)*:

— *Part 1: Hardware design requirements*

— *Part 2: Diagnostic protocol data unit application programming interface (D-PDU API)*

— *Part 3: Diagnostic server application programming interface (D-Server API)*

Introduction

The purpose of this part of ISO 22900 is to define a universal application programmer interface of a vehicle communication server application. At present, the automotive market requires different vehicle communication interfaces for different vehicle original equipment manufacturers (OEMs) supporting multiple communication protocols. However, up until now, many vehicle communication interfaces have been incompatible with regard to interoperability with multiple communication applications and vehicle communication protocols.

Implementation of the measurement calibration diagnostic (MCD) server concept supports overall cost reduction to the end user, e.g. because a single diagnostic or programming application will support many vehicle communication interfaces supporting different communication protocols and different vehicle communication modules of different vendors at one time.

A vehicle communication application compliant with this part of ISO 22900 supports a protocol-independent protocol data unit application programming interface (D-PDU API) as specified in ISO 22900-2. The server application needs to be configured with vehicle and electronic control unit (ECU) specific information. This is accomplished by supporting the open diagnostic exchange (ODX) data format, as specified in ISO 22901-1[1)].

1) Equivalent to ASAM MCD 2 D ODX[11].

Road vehicles — Modular vehicle communication interface (MVCI) —

Part 3: Diagnostic server application programming interface (D-Server API)

1 Scope

This part of ISO 22900 defines an object-oriented programming interface, the objective of which is to be able to implement server applications used during the design, production and maintenance phase of a vehicle communication system which are compatible with each other and therefore exchangeable.

A server compliant with this part of ISO 22900 has three function blocks:

— M: measurement,

— C: calibration, and

— D: diagnostics.

A more detailed description of the server API is given in the Programmers Reference Guide[14].

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

ISO 11898-2, *Road vehicles — Controller area network (CAN) — Part 2: High-speed medium access unit*

ISO 11898-3, *Road vehicles — Controller area network (CAN) — Part 3: Low-speed, fault-tolerant, medium-dependent interface*

ISO 11992-1, *Road vehicles — Interchange of digital information on electrical connections between towing and towed vehicles — Part 1: Physical and data-link layers*

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

ISO 14230-1, *Road vehicles — Diagnostic systems — Keyword Protocol 2000 — Part 1: Physical layer*

ISO 15765 (all parts), *Road vehicles — Diagnostics on Controller Area Networks (CAN)*

ISO 22900-2:2009, *Road vehicles — Modular vehicle communication interface (MVCI) — Part 2: Diagnostic protocol data unit application programming interface (D-PDU API)*

3 Terms, definitions, symbols and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1
AccessKey
path identifier through the inheritance hierarchy as defined in open diagnostic data exchange (ODX) to a diagnostic data element

3.1.2
ancestor object
parent object
object located above in the object hierarchy with respect to a given object

3.1.3
descendant object
child object
object located below in the object hierarchy with respect to a given object

3.1.4
functional class
set of diagnostic services

3.1.5
interface connector
connector at the vehicle end of the interface cable between the vehicle and the communication device

3.1.6
job
sequence of diagnostic services and other jobs with a control flow

3.1.7
location
representation of a set of diagnostic data valid on a given hierarchical level of inheritance as defined in open diagnostic data exchange (ODX)

EXAMPLE Multiple ECU Job; Protocol; Functional Group; ECU Base Variant; ECU Variant; Module.

3.1.8
logical link
set of data, identifying the physical line, the interface and protocol used for an electronic control unit (ECU)

3.1.9
physical interface link
physical connection between the vehicle communication interface (VCI) connector of a VCI and the interface connector

3.1.10
physical link
connection from the interface of the D-server to the electronic control unit (ECU) in the vehicle

NOTE A physical link is a **physical vehicle link** (3.1.11) connected to a **physical interface link** (3.1.9).

3.1.11
physical vehicle link
connection between the vehicle connector and the electronic control unit (ECU), which describes the unique bus system in a vehicle

3.1.12
project
pool of diagnostic, measurement and/or calibration data

NOTE References between such data are all resolvable within the same project.

3.1.13
vehicle connector
connector on a vehicle providing access to the bus systems in the vehicle

3.2 Abbreviated terms

A2L	ASAM 2 MC language (see ASAM MCD 2 MC[12])
API	Application Programming Interface
ASAM	Association for Standardisation of Automation and Measuring Systems
ASCII	American Standard for Character Information Interchange
AUSY	AUtomation SYstem
BVI	Base Variant Identification
BVIS	Base Variant Identification and Selection
C	Calibration
CAN	Controller Area Network
COM/DCOM	Distributed Component Object Model
CORBA	Common Object Request Broker Architecture
CRC	Cyclic Redundancy Check
D	Diagnostics
DDLID	Dynamically Defined Local ID
Diag	Diagnostic
DLL	Dynamic Link Library
DOP	diagnostic Data Object Property
D-server	Function block of MCD-server for Diagnostic
DTC	Diagnostic Trouble Code
DTD	Document Type Definition
DynId	Dynamic Identifiers
ECU	Electronic Control Unit
ECU MEM	Electronic Control Unit MEMory
ERD	Entity Relationship Diagram
IDL	Interface Definition Language
JAVA RMI	JAVA Remote Method Invocation
KWP	KeyWord Protocol
M	Measurement
MCD	Measurement Calibration Diagnostic
MC-server	Function block of MCD-server for Measurement and Calibration
MVCI	Modular Vehicle Communication Interface

ODX	Open Diagnostic data eXchange
OEM	Original Equipment Manufacturer
PC	Personal Computer
PDU	Protocol Data Unit (referred to as D-PDU API in this part of ISO 22900)
SDG	Special Data Groups
SI	Système International d'unités
UDS	Unified Diagnostic Services
UML	Unified Modeling Language
UTC	Coordinated Universal Time
VCI	Vehicle Communication Interface
VI	Variant Identification
VIS	Variant Identification and Selection

3.3 Typographical conventions and mnemonics used in this part of ISO 22900

Source code and technical artefacts within the text are presented in `Courier new` type.

Diagrams that denote interaction sequences, relationships or dependencies between interfaces are presented using the UML convention.

The name of each interface and each class defined by this part of ISO 22900 shall use the prefix of the stereotype, e.g. “MCD”.

The leading letter of each method and each parameter is small.

The leading word of each method shall be a verb.

The letter “_” is not permitted in interface names, method names and parameter names, but it is permitted for constants.

The leading letter of each constant is “e” followed by the name in block capitals.

ODX element names are written in upper case, e.g. SHORT-NAME.

Names of classes, objects and interfaces in ISO 22900-3 (MCD-3D) are written in mixed fixed, e.g. MCDDbProject.

3.4 Legends for used graphics

3.4.1 Hierarchical diagrams

Figure 1 illustrates the legends for hierarchical models.

	color print	black/white print
Interface not derived from MCDObject	blue	black
Interface not directly used	white	white
C data base interface	yellow	grey
C run time interface	green	dark grey
D data base interface	yellow	grey
D run time interface	green	dark grey
M data base interface	yellow	grey
M run time interface	green	dark grey
MC data base interface	yellow	grey
MC run time interface	green	dark grey
MD data base interface	yellow	grey
MD run time interface	green	dark grey
MCD data base interface	yellow	grey
MCD run time interface	green	dark grey

Figure 1 — Legend for hierarchical models

3.4.2 Sequence diagrams

With the help of sequence diagrams, the interactive use of the API and the sequences for certain general cases are presented in chronological order.

The sequence diagrams are oriented in accordance with the presentation in UML and are structured as follows. The chronological sequence arises while reading from the top downwards. The commentary column, in which single activities are commented on, is placed in the left margin. Within the sequence diagram, the Client application is shown left; if necessary for the respective case, the EventHandler is shown there as well. Right of the Client (with or without EventHandler) are located the API objects necessary for the respective case. On the far right the D-server is presented if necessary.

Not all API objects possible for the respective instant of time are shown, but only those of relevance for the respective case. The thin line leading down vertically from the objects represents the life line, the wider sections on it represent activities of the object.

The black horizontal arrows between the single objects, Client and D-server, represent the actions necessary for the respective case. The object at which the arrow points will execute the action. The grey horizontal arrows represent the return of objects.