
**Road vehicles — Modular vehicle
communication interface (MVCI) —**

Part 3:

**Diagnostic server application
programming interface (D-Server API)**

iTeh STANDARD PREVIEW
(standards.iteh.ai)

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

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

ISO 22900-3:2009

<https://standards.iteh.ai/catalog/standards/sist/aac796c8-0eb1-476e-b423-9dd0ac2f2fde/iso-22900-3-2009>



PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO 22900-3:2009

<https://standards.iteh.ai/catalog/standards/sist/aac796c8-0eb1-476e-b423-9dd0ac2f2fde/iso-22900-3-2009>



COPYRIGHT PROTECTED DOCUMENT

© ISO 2009

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
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword	viii
Introduction.....	ix
1 Scope	1
2 Normative references	1
3 Terms, definitions, symbols and abbreviated terms	2
3.1 Terms and definitions	2
3.2 Abbreviated terms	3
3.3 Typographical conventions and mnemonics used in this part of ISO 22900	4
3.4 Legends for used graphics	5
3.4.1 Hierarchical diagrams	5
3.4.2 Sequence diagrams	5
3.5 Stereotypes	6
4 General considerations	6
5 Specification release version information	6
6 Structure of MCD systems	7
7 Function block common MCD	10
7.1 MCD system object	10
7.2 Version information retrieval	11
7.3 Description of Terms	11
7.3.1 General	11
7.3.2 Client-controlled object	11
7.3.3 Location	11
7.3.4 Logical Link (LOGICAL-LINK)	12
7.3.5 Project	12
7.3.6 Server-controlled mutable object	13
7.3.7 Server-controlled object (shared object)	13
7.4 States of the MCD system object	13
7.5 State changes	15
7.6 Project configuration	16
7.7 Interface structure of MCD-server API	18
7.7.1 Separation in database and runtime side	18
7.7.2 Hierarchical model	19
7.8 Structure of the database	20
7.8.1 Overview	20
7.8.2 Associations of DbLocation for MCD	20
7.8.3 Database within the field Measurement and Calibration	22
7.8.4 Database within the field Diagnostics	22
7.9 Collections	22
7.9.1 Types and methods	22
7.9.2 RunTime collections	24
7.9.3 Database collections	26
7.9.4 Handling of collection of ASCIISTRING	28
7.10 EventHandler	28
7.10.1 Registering/deregistering of the EventHandlers	28
7.10.2 Methods of the EventHandlers	30
7.10.3 Eventfilter	35
7.11 Multi-Client capability	43
7.11.1 Requirements	43

7.11.2	Design	46
7.11.3	Proxy in Multi Client Architecture	47
7.11.4	Cooperation Level.....	50
7.11.5	Symbolic Names of Clients.....	52
7.11.6	Selection and de-selection of Project and VehicleInfo in a multi-client setting	52
7.11.7	Notification	53
7.11.8	Remove shared objects	54
7.11.9	Locking	55
7.12	Client Controlled Objects.....	57
7.13	Resource Release	58
7.13.1	Use cases	58
7.13.2	Requirements	59
7.13.3	Solution.....	59
7.14	Critical Section, Critical Groups of Methods	59
7.15	Result access	61
7.16	MCD value.....	62
7.16.1	Value types	62
7.16.2	Method getValue	63
7.16.3	Method setValue	63
7.16.4	Method createValue	64
7.17	Use cases	68
7.17.1	View	68
7.17.2	Instantiation of projects	68
7.17.3	Database access	71
7.17.4	Destruction	73
8	Function block Common MD.....	74
8.1	Collector	74
8.1.1	ERD	74
8.1.2	Concept.....	75
8.1.3	Result access	79
8.1.4	Collector usage in diagnostics	84
8.2	Use cases	85
8.2.1	Measurement with a collector - activation	85
8.2.2	Measurement with a collector - result access	87
8.2.3	Measurement with a collector - polling for results.....	90
9	Function block Diagnostics.....	91
9.1	Description of Terms	91
9.1.1	General.....	91
9.1.2	Access Key.....	91
9.1.3	Functional Class (FUNCTIONAL-CLASS).....	91
9.1.4	Job (SINGLE-ECU-JOB, MULTIPLE-ECU-JOB).....	91
9.1.5	Physical Interface Link.....	91
9.1.6	Physical Link	92
9.1.7	Physical Vehicle Link (PHYSICAL-VEHICLE-LINK)	92
9.2	Structuring of the function block Diagnostics	92
9.2.1	Separation in database and runtime side	92
9.2.2	Relation between Vehicle Connector Information Table and Logical Link Table	94
9.2.3	Hierarchical model.....	95
9.2.4	Entity Relationship Diagrams	97
9.3	System Properties	109
9.4	Diagnostic DiagComPrimitives and Services	109
9.4.1	Diagnostic DiagComPrimitives and States	109
9.4.2	Service overview.....	113
9.4.3	Non cyclic single shot diag service	120
9.4.4	Cyclic diag service.....	122
9.4.5	Repeated diag service.....	123
9.4.6	Repeated send only diag service	124
9.4.7	Repeated receive only diag service	125
9.4.8	Updating repetition parameters	125

STANDARD PREVIEW

(standards.iteh.ai)

ISO 22900-3:2009

https://standards.iteh.ai/catalog/standards/sist/aac796c8-0cb1-476c-b423-

9dd0ac2f2fde/iso-22900-3-2009

9.4.9	Summary	126
9.4.10	Protocol parameters.....	127
9.4.11	Suppress Positive Response	140
9.5	Diagnostic variables.....	143
9.6	eEND_OF_PDU as RequestParameter	145
9.6.1	Database side	145
9.6.2	Runtime side	147
9.6.3	COMPUCODE	148
9.7	Variable length parameters	149
9.8	Layer inheritance of services	151
9.8.1	Goal.....	151
9.8.2	Layer inheritance of services	151
9.8.3	Service handling on functional and physical locations	156
9.9	Variant Identification and Selection (VI / VIS).....	158
9.9.1	Goal.....	158
9.9.2	Variant Identification Algorithm	158
9.9.3	General VI/VIS handling considerations	166
9.9.4	Deselecting of selected variants.....	168
9.9.5	Request and Response parameters of VI and VIS	168
9.9.6	Example Scenarios for VI and VIS	173
9.10	Base Variant Identification and Selection.....	183
9.11	Use Cases	190
9.11.1	Creation of LogicalLink and usage of DiagComPrimitives	190
9.11.2	Removal of communication objects	192
9.11.3	Service Handling	194
9.11.4	Result access	196
9.12	Filtering of results	225
9.12.1	Principle.....	225
9.12.2	Handling rules.....	229
9.13	Read DTC.....	230
9.14	Logical Link.....	234
9.14.1	General	234
9.14.2	Connection overview	234
9.14.3	State diagram of Logical Link	235
9.14.4	Logical Link examples	243
9.14.5	Gateway handling.....	248
9.14.6	Examples and Relations between Logical Links, Locations and relevant AccessKeys.....	250
9.15	Functional Addressing.....	256
9.16	Tables	258
9.16.1	General	258
9.16.2	Usage of tables within DiagComPrimitives	260
9.17	Dynamically Defined Local Id / Table Parameters (DDLID)	263
9.17.1	General	263
9.17.2	DDLID principle and requirements	263
9.17.3	Lifecycle	264
9.18	Internationalisation	272
9.18.1	Multi language support.....	272
9.18.2	Units.....	272
9.19	Special Data Groups	272
9.20	ECU Flash programming	273
9.20.1	Goal.....	273
9.20.2	Description of Terms for ECU-Reprogramming	274
9.20.3	Structure of the function block flash programming	275
9.20.4	Management of ECU-MEMs	288
9.20.5	Physical Memories	289
9.20.6	Executing flash sessions.....	291
9.21	Library.....	299
9.22	Java Jobs	300
9.22.1	General	300
9.22.2	General information Java Jobs.....	300

9.22.3	Types of Java Jobs.....	301
9.22.4	Handling of Java Jobs.....	304
9.22.5	Job execution.....	314
9.22.6	Job example.....	322
9.23	ECU configuration.....	332
9.23.1	General.....	332
9.23.2	ECU Configuration Database Part.....	332
9.23.3	ECU Configuration Runtime Part.....	336
9.23.4	Error Handling.....	339
9.23.5	Initialising an MCDConfigurationRecord.....	339
9.23.6	Offline versus Online Configuration.....	340
9.23.7	Uploading and Downloading Configuration Strings.....	341
9.24	Audiences and Additional Audiences.....	346
9.24.1	General.....	346
9.24.2	Audiences.....	348
9.24.3	Additional Audiences.....	348
9.25	Function Dictionary and Sub-Components.....	349
9.25.1	Terms and requirements.....	349
9.25.2	Functions and function group in ODX.....	349
9.25.3	Function dictionary data model description.....	351
9.25.4	Function dictionary usage scenario.....	353
9.25.5	Sub-Component data model description.....	355
9.25.6	Sub-Component usage scenario.....	356
9.26	ECU States.....	357
9.27	Monitoring vehicle bus traffic.....	360
9.28	Support of VCI module selection and other VCI module features in accordance with D-PDU API Standard.....	362
9.28.1	General.....	362
9.28.2	Definitions.....	362
9.28.3	General behaviour of D-PDU API related D-server methods.....	362
9.28.4	Overview of VCI module related classes.....	363
9.28.5	VCI module selection.....	364
9.28.6	MCDInterface.....	364
9.28.7	VCI module selection sequence.....	365
9.28.8	Interface status events.....	366
9.28.9	MCDInterfaceResource.....	367
9.28.10	Selection of an interface resource.....	367
9.28.11	Send Break Signal.....	368
9.28.12	MCDDbInterfaceCable.....	369
9.28.13	Accessing VCI module features.....	369
9.28.14	Adding Logical Links which are not found in the Vehicle Information.....	370
9.28.15	Behaviour of a MCD-server not supporting VCI Modules in accordance with D-PDU API Standard.....	371
9.29	Mapping of D-PDU API methods.....	372
9.29.1	General.....	372
9.29.2	Initialization and Selection of VCI Modules.....	372
9.29.3	Communication on a Logical Link.....	372
9.29.4	Handling of Communication Parameters.....	374
9.29.5	MCDStartCommunication and MCDStopCommunication.....	376
10	Error Codes.....	377
10.1	Principle.....	377
10.2	Description of the errors.....	379
10.2.1	Error free behaviour.....	379
10.2.2	Parameterisation errors.....	379
10.2.3	Runtime / ProgramViolation errors.....	379
10.2.4	Database errors.....	379
10.2.5	System errors.....	379
10.2.6	Communication errors.....	379
10.2.7	Job error.....	379

Annex A (informative) Code examples	380
Annex B (normative) COMPUCODE template	387
Annex C (normative) Job Templates	388
Annex D (informative) Gateway Handling	392
Annex E (normative) Value reading and setting by string	393
Annex F (normative) Bus types	396
Annex G (normative) System parameter	398
Annex H (normative) Cooperation level	401
Annex I (normative) System properties	428
Annex J (informative) VCI Module selection sequence	431
Annex K (informative) Service example illustrations	432
Annex L (informative) Exemplary monitoring message format	437
Annex M (normative) Overview of the methods for isModifiedByOtherClient flag	439
Bibliography	447

iTeh STANDARD PREVIEW (standards.iteh.ai)

[ISO 22900-3:2009](https://standards.iteh.ai/catalog/standards/sist/aac796c8-0eb1-476e-b423-9dd0ac2f2fde/iso-22900-3-2009)

<https://standards.iteh.ai/catalog/standards/sist/aac796c8-0eb1-476e-b423-9dd0ac2f2fde/iso-22900-3-2009>

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* [ISO 22900-3:2009](https://standards.iteh.ai/catalog/standards/sist/aac796c8-0eb1-476e-b423-9dd0ac2121de/iso-22900-3-2009)
- *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 22901-1¹⁾.

iTeh STANDARD PREVIEW (standards.iteh.ai)

[ISO 22900-3:2009](https://standards.iteh.ai/catalog/standards/sist/aac796c8-0eb1-476e-b423-9dd0ac2f2fde/iso-22900-3-2009)

<https://standards.iteh.ai/catalog/standards/sist/aac796c8-0eb1-476e-b423-9dd0ac2f2fde/iso-22900-3-2009>

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

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO 22900-3:2009

<https://standards.iteh.ai/catalog/standards/sist/aac796c8-0eb1-476e-b423-9dd0ac2f2fde/iso-22900-3-2009>

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.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO 22900-3:2009](https://standards.iteh.ai/catalog/standards/sist/aac796c8-0eb1-476e-b423-100000000000/iso-22900-3-2009)

[https://standards.iteh.ai/catalog/standards/sist/aac796c8-0eb1-476e-b423-](https://standards.iteh.ai/catalog/standards/sist/aac796c8-0eb1-476e-b423-100000000000/iso-22900-3-2009)

A more detailed description of the server APIs 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

ISO 22900-3:2009(E)

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 MCDOobject	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.