

Second edition
2012-12-01

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 22900-3:2012(E)

© ISO 2012

This is a preview of "ISO 22900-3:2012". [Click here to purchase the full version from the ANSI store.](#)



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
Web www.iso.org

Published in Switzerland

This is a preview of "ISO 22900-3:2012". Click here to purchase the full version from the ANSI store.

Contents

Page

Foreword	v
Introduction.....	vi
1 Scope	1
2 Normative references.....	1
3 Terms, definitions, symbols and abbreviated terms	1
3.1 Terms and definitions	1
3.2 Symbols.....	3
3.3 Abbreviated terms	4
4 Conventions	5
4.1 General	5
4.2 Typographical conventions and mnemonics	5
4.3 Sequence diagrams.....	6
4.4 Stereotypes	6
5 Specification release version information	6
6 Structure of a MVCI diagnostic server	6
7 Diagnostic server	10
7.1 MCD system object	10
7.2 Description of terms.....	11
7.3 Version information retrieval	16
7.4 States of the MCD system	16
7.5 State changes	19
7.6 Project configuration	19
7.7 Interface structure of server API.....	21
7.8 Collections	46
7.9 Registering/deregistering of the EventHandler.....	50
7.10 MCD value	51
7.11 Use cases	54
8 Function block Diagnostic in detail.....	60
8.1 Constraints.....	60
8.2 System Properties	70
8.3 Diagnostic DiagComPrimitives and Services.....	71
8.4 Suppress positive response	101
8.5 eEND_OF_PDU as RequestParameter	102
8.6 Variable length parameters	104
8.7 Variant identification	106
8.8 Use cases	117
8.9 Read DTC	135
8.10 Logical Link.....	144
8.11 Functional addressing	156
8.12 Tables	158
8.13 Dynamically Defined Identifiers (DynId).....	168
8.14 Internationalization.....	179
8.15 Special Data Groups	179
8.16 ECU (re-) programming.....	181
8.17 Handling binary flash data	188
8.18 Library.....	190
8.19 Jobs	191
8.20 ECU configuration	212

This is a preview of "ISO 22900-3:2012". Click here to purchase the full version from the ANSI store.

8.21	Audiences and additional audiences	229
8.22	ECU states	231
8.23	Function dictionary	234
8.24	Sub-Component data model description	242
8.25	Monitoring vehicle bus traffic.....	244
8.26	Support of VCI module selection and other VCI module features according to ISO 22900-2 ..	246
8.27	Handling DoIP entities.....	255
8.28	Mapping of D-PDU API methods	258
9	Error Codes	263
9.1	Principle.....	263
9.2	Description of the errors.....	265
Annex A	(normative) Value reading and setting by string.....	267
A.1	Datatype conversion into Unicode2 string	267
A.2	Representation floating numbers	267
A.3	Normalized floating-point numbers	268
Annex B	(normative) System parameter.....	269
B.1	Overview	269
B.2	Description of the system parameters	270
Annex C	(normative) Overview optional functionalities	272
Annex D	(informative) Monitoring message format.....	278
D.1	General.....	278
D.2	CAN format.....	278
D.3	K-Line Format.....	279
D.4	DoIP Format.....	280
Bibliography	281

This is a preview of "ISO 22900-3:2012". [Click here to purchase the full version from the ANSI store.](#)

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*.

This second edition cancels and replaces the first edition (ISO 22900-3:2009), which has been technically revised.

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

0.1 Overview

This part of ISO 22900 has been established in order to define a universal application programmer interface of a vehicle communication server application. Today's situation in the automotive market requires different vehicle communication interfaces for different vehicle OEMs supporting multiple communication protocols. However, until today, many vehicle communication interfaces are incompatible with regard to interoperability with multiple communication applications and vehicle communication protocols.

Implementation of the MVCI diagnostic server concept supports overall cost reduction to the end user because, for example, 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 D-PDU API (Protocol Data Unit Application Programming Interface) as specified in ISO 22900-2. The server application will need to be configured with vehicle- and ECU-specific information. This is accomplished by supporting the ODX data format (Open Diagnostic Exchange format) as specified in ISO 22901-1.

A server compliant with this part of ISO 22900 supports the function block Diagnostics (D). A compliant server also supports Job-Language (Java) and may support optional features like ECU (re)programming. The defined object-oriented API provides for a simple, time saving and efficient interchangeability of different servers.

The client application and the communication server do not necessarily need to run on the same computer. A remote use via an interface may also be envisaged and is supported by the design of the server API. This interface is provided for ASAM GDI, COM/DCOM ^[10] [Technology Reference COM-IDL], for C++ ^[11] [Technology Reference C++] and for Java ^[12] [Technology Reference Java].

0.2 ASAM e.V. implementation reference documents

This part of ISO 22900 references several ASAM e.V. documents which contain the Technology Reference Mapping Rules for COM-IDL, C++ and Java.

The following ASAM documents are relevant for the implementation of this part of ISO 22900:

- ASAM Technology Reference COM-IDL, *COM-IDL Technology Reference Mapping Rules* ^[10]:
this document describes the platform, programming language and linking mechanisms for the implementation of the generic object model in COM-IDL.
- ASAM Technology Reference C++, *C++ Technology Reference Mapping Rules* ^[11]:
this document describes the platform, programming language and linking mechanisms for the implementation of the generic object model in C++.
- ASAM Technology Reference Java, *Java Technology Reference Mapping Rules* ^[12]:
this document describes the platform, programming language and linking mechanisms for the implementation of the generic object model in Java.