International Standard

ISO 17978-3

First edition
2026-03

Road vehicles — Service-oriented vehicle diagnostics (SOVD) —

Part 3: Application programming interface (API)

Véhicules routiers — Diagnostic Véhicule Orienté Services (SOVD) —

Partie 3: Interface de Programmation D'applications (API)

Reference number
ISO 17978-3:2026(en)

© ISO 2026



COPYRIGHT PROTECTED DOCUMENT

© ISO 2026

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
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword ..... **vii**
Introduction ..... **viii**
1 Scope **1**
2 Normative references **1**
3 Terms and definitions **2**
4 Symbols and abbreviated terms **3**
4.1 Symbols 3
4.2 Abbreviated terms 3
5 SOVD API general aspects **5**
5.1 Use cases overview 5
5.2 Accessing the API 5
5.2.1 HTTP REST based approach 5
5.2.2 Server-sent events 7
5.3 Entities 9
5.3.1 General 9
5.3.2 EntityTypes and collections 10
5.3.3 Entity hierarchy and relations 11
5.3.4 Entity identifier and path 13
5.3.5 Location of the SOVD server 13
5.3.6 Multi-SOVD server environments 13
5.4 Resources 14
5.4.1 General 14
5.4.2 Standardized resource collections 14
5.4.3 Standardized resources 15
5.4.4 Restricting access to entities and resources by authentication 16
5.4.5 Custom specific resource collections and types 16
5.5 Illustration of SOVD concepts for entities and resources 17
5.6 SOVD API versioning 18
5.7 Internationalization and localization 18
5.7.1 Internationalization 18
5.7.2 Unit definitions 18
5.7.3 Internationalization of enumerations 20
5.7.4 Other localization topics (numbers and dates) 20
5.8 Response status codes 21
5.8.1 General 21
5.8.2 Generic definition of status codes 21
5.8.3 Definition of error details 22
5.8.4 SOVD error codes for common cases 22
5.9 Common primitive data types 23
5.10 Definition of JSON schema and objects for vehicle responses 24
5.11 Discovery of the SOVD server for the co-located use case cluster 26
5.11.1 General concept 26
5.11.2 Network connection 26
5.11.3 Address an SOVD server using mDNS 26
5.11.4 Discovery of the SOVD server using DNS-SD 26
5.11.5 Example 27
5.12 Discovery of the SOVD server for the remote use case cluster 27
6 Capability description **27**
6.1 General 27
6.2 Capability description structure 28
6.2.1 General 28
6.2.2 Info object 28
6.2.3 Path item object 28

© ISO 2026 – All rights reserved

6.2.4 Operation object ... 30
6.2.5 Schema object ... 30
6.2.6 Header object ... 31
6.2.7 Predefined query parameter ... 31
6.2.8 `x-sovd-tags` object ... 32
6.3 Online and offline capability descriptions ... 33
6.3.1 General ... 33
6.3.2 Offline capability description ... 33
6.3.3 Online capability description ... 33
6.4 Variant handling in offline capability description ... 33
6.4.1 General ... 33
6.4.2 Extension: x-sovd-applicability ... 34
6.4.3 Usage of `x-sovd-applicability` by the SOVD client ... 35

7 SOVD REST API ... 36
7.1 General ... 36
7.2 Definition of conventions for API methods ... 36
7.2.1 Query parameters ... 36
7.2.2 Request body ... 36
7.3 Timeout definitions ... 36
7.4 API methods for access of supported SOVD server versions ... 37
7.4.1 General ... 37
7.4.2 Read all supported versions of the SOVD server ... 37
7.5 API methods for access to capability description content ... 38
7.5.1 General ... 38
7.5.2 Retrieve online capability description ... 39
7.6 API methods for discovery of Entities and resources ... 40
7.6.1 General ... 40
7.6.2 Query for Entities from an SOVD server ... 40
7.6.3 Query for capabilities of an entity ... 46
7.7 Enhancement of API methods for tagged Entities and resources ... 49
7.7.1 General ... 49
7.7.2 Query for tagged Entities and resources ... 49
7.7.3 Retrieve `tags` from entity or a resource ... 50
7.7.4 Query for tagged capabilities of an entity ... 52
7.8 API methods for fault handling ... 53
7.8.1 General ... 53
7.8.2 Query for faults from an entity ... 53
7.8.3 Read a fault resource ... 57
7.8.4 Delete all faults of an entity ... 59
7.8.5 Delete single fault of an entity ... 60
7.9 API methods for data resource read/write access ... 61
7.9.1 General ... 61
7.9.2 Retrieve of data resource categories and groups of an entity ... 62
7.9.3 Query for data from an entity ... 65
7.9.4 Read a data resource ... 67
7.9.5 Read multiple data values from an entity ... 70
7.9.6 Write a data resource ... 75
7.10 API methods for cyclic-subscriptions ... 77
7.10.1 General ... 77
7.10.2 Query for cyclic-subscriptions from an entity ... 77
7.10.3 Issue a new cyclic-subscription ... 78
7.10.4 Retrieve details of a cyclic-subscription ... 82
7.10.5 Update a cyclic-subscription ... 83
7.10.6 Delete a cyclic-subscription ... 84
7.11 API methods for triggers ... 85
7.11.1 General ... 85
7.11.2 Issue a new trigger ... 87
7.11.3 Query for triggers from an entity ... 93

© ISO 2026 – All rights reserved

7.11.4 Retrieve details of a trigger ... 95
7.11.5 Update a trigger ... 96
7.11.6 Delete a trigger ... 98
7.11.7 Trigger value reception ... 99
7.12 API methods for configuration ... 100
7.12.1 General ... 100
7.12.2 Query for configurations from an entity ... 100
7.12.3 Read a configuration resource ... 102
7.12.4 Write a configuration resource ... 105
7.12.5 Reset configuration ... 108
7.13 API methods for clearing data ... 110
7.13.1 General ... 110
7.13.2 Retrieve list of supported clear-data types ... 111
7.13.3 Start clearing of data ... 112
7.13.4 Retrieve status of clearing data ... 114
7.14 API methods for control of operations ... 116
7.14.1 General ... 116
7.14.2 Ensuring co-located diagnostics ... 116
7.14.3 Query for operations from an entity ... 116
7.14.4 Query for executions of an operation ... 118
7.14.5 Retrieve details of an operation ... 119
7.14.6 Start execution of an operation ... 122
7.14.7 Read the status of an operation execution ... 125
7.14.8 Terminate the execution of an operation ... 126
7.14.9 Support for execute / freeze / reset and custom capabilities ... 127
7.15 API methods for script execution ... 129
7.15.1 General ... 129
7.15.2 Upload a script ... 130
7.15.3 Query for scripts from an entity ... 132
7.15.4 Retrieve details of a script ... 133
7.15.5 Remove a script ... 135
7.15.6 Start execution of a script ... 136
7.15.7 Query for executions of a Script ... 139
7.15.8 Read the status of a script execution ... 140
7.15.9 Terminate the script execution ... 141
7.15.10 Remove the execution of a script ... 143
7.16 API methods for support of target modes ... 144
7.16.1 General ... 144
7.16.2 Query for modes of an entity ... 144
7.16.3 Retrieve details of a mode ... 146
7.16.4 Explicit control of entity states via their defined modes ... 147
7.16.5 Hint-based control of entity states via their defined modes ... 149
7.17 API methods for locking ... 149
7.17.1 General ... 149
7.17.2 Acquire a lock on an entity ... 152
7.17.3 Query for locks of an entity ... 154
7.17.4 Retrieve details of a lock ... 155
7.17.5 Modify the expiration time of a lock on an entity ... 156
7.17.6 Release an acquired lock on an entity ... 158
7.18 API methods for software update ... 159
7.18.1 General ... 159
7.18.2 Query for updates from an entity ... 161
7.18.3 Retrieve details of an update ... 163
7.18.4 Automated installation of an update ... 165
7.18.5 Prepare installation of an update ... 166
7.18.6 Execute installation of an update ... 168
7.18.7 Read the status of an update ... 170
7.18.8 Delete update package from an SOVD server ... 172
7.18.9 Register an update at the SOVD server ... 173

© ISO 2026 – All rights reserved

7.19 API methods for restarting an entity ... 174
7.19.1 General ... 174
7.19.2 Read the status of an entity ... 175
7.19.3 Start an entity ... 176
7.19.4 Restart an entity ... 178
7.19.5 Forced Restart an entity ... 179
7.19.6 Shutdown an entity ... 180
7.19.7 Forced Shutdown an entity ... 181
7.20 API methods for handling of bulk-data ... 183
7.20.1 General ... 183
7.20.2 Retrieve bulk-data categories of an entity ... 184
7.20.3 Retrieve details of bulk-data resource ... 186
7.20.4 Download and upload bulk-data ... 188
7.20.5 Delete bulk-data ... 192
7.21 API methods for logging ... 194
7.21.1 General ... 194
7.21.2 Retrieve resources of an SOVD log ... 195
7.21.3 Read log entries from an SOVD log ... 196
7.21.4 Configure SOVD log ... 199
7.21.5 Retrieve the current SOVD log configuration from an entity ... 200
7.21.6 Reset SOVD logging configuration to default ... 202
7.22 API methods for communication-logs ... 203
7.22.1 General ... 203
7.22.2 Query for communication-logs from an entity ... 203
7.22.3 Configure communication-logs for an entity ... 204
7.22.4 Read the status of a communication-log ... 206
7.22.5 Control a communication-log ... 207
7.22.6 Delete communication-log ... 208
7.23 Authentication of SOVD clients (informative) ... 210
7.23.1 Online and offline authentication and authorization ... 210
7.23.2 Securing connectionc ... 215
7.23.3 Verifying SOVD client credentials and requesting a token at the backend ... 215
7.23.4 Verifying SOVD client credentials ... 215
7.23.5 Requesting a token ... 216
7.23.6 Request header for access-restricted resources ... 217
7.23.7 Validating a token ... 218

8 Classic diagnostic adapter ... 218
8.1 General ... 218
8.2 Access to UDS based entities ... 218
8.3 Specific mapping of UDS services to SOVD modes ... 219
8.3.1 Mapping of DiagnosticSessionControl (10_{16}) subfunctions to SOVD modes ... 219
8.3.2 Mapping of SecurityAccess (27_{16}) subfunctions to SOVD modes ... 220
8.3.3 Mapping of Authentication (29_{16}) to SOVD modes ... 220
8.3.4 Mapping of CommunicationControl (28_{16}) subfunctions to SOVD modes ... 220
8.3.5 Mapping of ControlDTC Setting (85_{16}) subfunctions to SOVD modes ... 220
8.4 Mapping of UDS services to data resources ... 220
8.5 Mapping of UDS services to fault resources ... 221
8.6 Mapping of UDS services to operation resources ... 221
8.6.1 Mapping of Routine (31_{16}) and InputOutputControlByIdentifier ($2F_{16}$) services to operation resources ... 221
8.6.2 DEPRECATED — Mapping of EcuReset service (11_{16}) to operation resources ... 221
8.7 Mapping of EcuReset service (11_{16}) to restart an entity resource ... 221

Annex A (normative) **Access to legally required diagnostics ... 222**

Annex B (informative) **SOVD API common example ... 225**

Bibliography ... 235

© ISO 2026 – All rights reserved

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 document 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).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

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.

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

A list of all parts in the ISO 17978 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.

© ISO 2026 – All rights reserved

Introduction

The introduction of high-performance computers (HPCs) in vehicles is associated with changes in the traditional electronic and electrical vehicle architectures. Besides classical distributed embedded electronic control unit (ECU) architectures, domain- or zone-based architectures are also available. These architectures also extend beyond the physical vehicle.

This extends the focus from checking hardware to also checking the functionality of applications. This includes the recording of data such as memory usage, processor load, and the number of active services, as well as the collection of log and trace files.

These topics result in challenges regarding the management of the vehicle life cycle. Further aspects are:

— faster release and update cycles;

— increased requirements such as data protection and cybersecurity;

— state-of-the-art diagnostic API using current information technologies.

The ISO 17978 series defines an API which standardizes the methods for:

— discovering the SOVD capabilities;

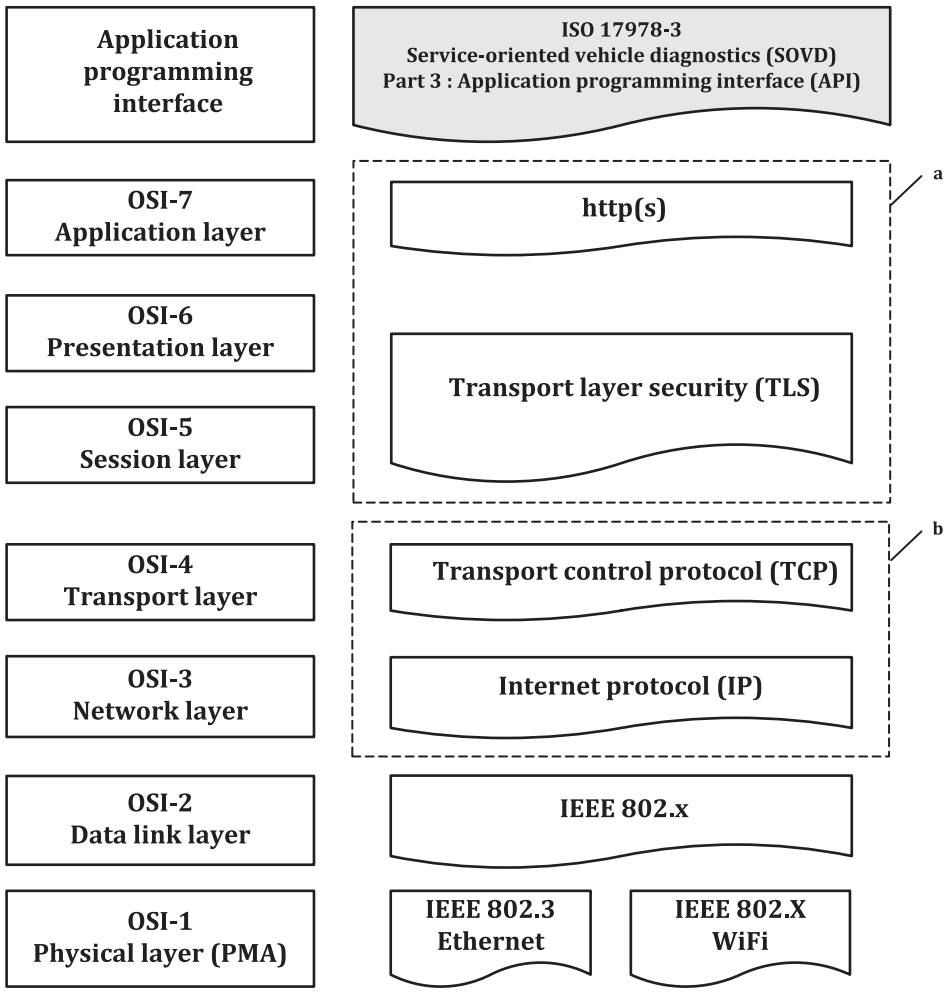
— performing diagnostics;

— (re-)configuring and re-programming;

— allowing the introduction of new functionalities.

Figure 1 shows the OSI layers of the ISO 17978 series.

© ISO 2026 – All rights reserved



a Communication protocol.

b Network technology depending on electric/electronic (E/E) vehicle network architecture.

Figure 1 — OSI layers of SOVD

For the convenience of the user, a machine-readable OpenAPI definition of the methods is published alongside this document in the ISO Standards Maintenance Portal: https://standards.iso.org/iso/17978/-3/ed-1/en.

NOTE This document marks some API methods as deprecated. The background to this is the release of ASAM SOVD version 1.0 in calendar year 2022.

© ISO 2026 – All rights reserved

Road vehicles — Service-oriented vehicle diagnostics (SOVD) —

Part 3: Application programming interface (API)

1 Scope

This document defines an application programming interface (API) which standardizes the methods for diagnosing HPCs and legacy ECUs, the retrieval of the diagnostic capabilities, and the discovery of the service-oriented vehicle diagnostics (SOVD) methods in an extended vehicle. The SOVD API provides a unified access to classic ECUs and HPCs.

The SOVD API provides functions such as:

— access to the faults, including reading the fault entries, reading environment data, and deleting fault entries;

— measurements and identifications of all entities;

— execution of routines, input/output (I/O) controls and software functions in certain modes or states;

— configuration of a vehicle (e.g. equipment, country, customer demand, variant coding);

— encapsulation of the ExVe manufacturer specific software update strategy (including firmware over the air (FOTA) if available);

— access to logging information of an HPC.

With these features, SOVD can cover all areas of the vehicle life cycle, including engineering (development), manufacturing (production), after sales (maintenance and repair), legal and technical inspections and vehicle operation (use).

There are several aspects which are not covered by this document, as they are specific to the implementation of an SOVD server, for example:

— prevention and quick reaction to attack vectors like denial of service, zero-day exploits of vulnerabilities;

— recognition of security incidents;

— maintaining the operational safety based on data monitoring;

— management of security incidents;

— data privacy issues;

— load balancing of concurrent requests.

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 2026 – All rights reserved

ISO 8601-1, *Date and time — Representations for information interchange — Part 1: Basic rules*

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

ECMA 262: 16, *th edition, ECMAScript general-purpose programming language*[1)]

REST. Architectural Styles and the Design of Network-based Software Architectures, *PhD dissertation Fielding R.T., University of California, Irvine,2000*[2)]

HTML Server-Sent Events API[3)]

IETF RFC 1341, *MIME (Multipurpose Internet Mail Extensions)*

IETF RFC 1847:1995, *Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted*

IETF RFC 3339:2002, *Date and Time on the Internet: Timestamps*

IETF RFC 3986:2005, *Uniform Resource Identifier (URI): Generic Syntax*

IETF RFC 5322:2008, *Internet Message Format*

IETF RFC 6570:2012, *URI Template*

IETF RFC 6762:2013, *Multicast DNS*

IETF RFC 6763:2013, *DNS-Based Service Discovery*

IETF RFC 7540:2015, *Hypertext Transfer Protocol Version 2 (HTTP/2)*

IETF RFC 8259:2017, *The JavaScript Object Notation (JSON) Data Interchange Format*

IETF RFC 9110:2022, *HTTP Semantics*

IETF RFC 9562:2024, *Universally Unique IDentifiers (UUIDs)*

NodeJS, *Semantic Versioning Range Definition*

Open API Initiative, OpenAPI Specification v3.1.0; 2021[4)]

OpenJS Foundation, *JSON Schema 2020-12 Release Notes*

Semantic, *Versioning 2.0.0*

3 Terms and definitions

For the purposes of this document, the following terms and definitions 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/

3.1
application
app
entity (3.5) representing a software program that performs a specific task

1) Available at: https://tc39.es/ecma262/

2) Available at: https://ics.uci.edu/~fielding/pubs/dissertation/top.htm

3) Available at: https://www.w3schools.com/html/html5_serversentevents.asp

4) https://standards.iso.org/iso/17978/-3/

© ISO 2026 – All rights reserved

3.2
area
entity (3.5) representing a logical view on vehicle topologies

3.3
ContributingEntity
abstract *entity* (3.5) that describes relationships like hosts, is-located-on, contains, depends-on, belongs-to to represent dependencies between specialized entities *app* (3.1), *area* (3.2), *component* (3.4), and *function* (3.6)

3.4
component
entity (3.5) representing hardware (characterized by the availability of a central processing unit) or software (characterized by the availability of an operating system or hypervisor), or a combination of both

3.5
entity
object combining logically interrelated diagnostic information represented in the form of resources

Note 1 to entry: *entity* (3.5) is an abstract base class. The specific entities describe a diagnostic view of the vehicle.

Note 2 to entry: The term *entity* (3.5) used in the context of this document refers to "SOVD entity" (see ISO 17978-2).

3.6
function
entity (3.5) representing a functional view on vehicle information spread across *components* (3.4)

3.7
SOVDServer
entity (3.5) representing the server-side implementation of the representational state transfer (REST) API services

4 Symbols and abbreviated terms

4.1 Symbols

—	empty table cell

4.2 Abbreviated terms

AI	artificial intelligence
App	application
AUTOSAR_DLT	AUTOSAR Specification of Diagnostic Log and Trace
C	conditional
CDA	classic diagnostic adapter
CPU	central processing unit
DHCP	dynamic host configuration protocol
DLT	diagnostic log and trace
DNS	domain name system
DNS-SD	DNS service discovery

© ISO 2026 – All rights reserved