



BSI Standards Publication

Intelligent transport systems — Field device Simple Network Management Protocol (SNMP) data interface

Part 1: Global objects

National foreword

This Published Document is the UK implementation of ISO/TS 26048-1:2025.

The UK participation in its preparation was entrusted to Technical Committee EPL/278, Intelligent transport systems.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

This publication is not to be regarded as a British Standard.

© The British Standards Institution 2025
Published by BSI Standards Limited 2025

ISBN 978 0 539 29457 6

ICS 35.240.60

Compliance with a Published Document cannot confer immunity from legal obligations.

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 31 July 2025.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------



Technical Specification

ISO/TS 26048-1

Intelligent transport systems — Field device Simple Network Management Protocol (SNMP) data interface —

Part 1: Global objects

*Systèmes de transports intelligents — Interface de données
Simple Network Management Protocol (SNMP) pour les
dispositifs de terrain —*

Partie 1: objets globaux

**First edition
2025-07**



COPYRIGHT PROTECTED DOCUMENT

© ISO 2025

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
1 Scope	1
2 Normative references	1
3 Terms and definitions	2
4 Symbols and abbreviated terms	5
5 Conventions and architecture	6
5.1 ISO maintenance portal	6
5.2 MIB files	6
5.2.1 MIB format	6
5.2.2 MIB access	7
5.2.3 MIB filenames	7
5.3 ASN.1	7
5.4 Conformance	7
5.5 SNMP terminology	7
5.6 Architecture	8
5.6.1 ITS services	8
5.6.2 Functional view of interface	8
5.6.3 Physical view	8
5.6.4 Communications view	8
5.7 Constraints	9
5.7.1 Table design constraint	9
5.7.2 Access constraints	10
5.8 Outline	10
5.9 Requirements pattern	10
6 User needs	12
6.1 Authenticate users	12
6.2 Control access to data	12
6.3 Monitor failed access to the field device	12
6.4 Manage the field device	12
6.5 Monitor field device components	12
6.5.1 Monitor ambient environment	12
6.5.2 Monitor field device enclosure	13
6.5.3 Manage enclosure climate control components	13
6.5.4 Monitor field device power	14
6.5.5 Manage auxiliary ports	14
6.6 Receive notification of triggers firing	14
6.7 Manage device-specific notifications	15
6.8 Log system events	15
6.9 Log user-defined data snapshots	15
6.10 Record a series of data snapshots	15
6.11 Issue trigger-based commands	15
6.12 Configure a complex device	16
6.13 Efficient exchange of data	16
6.14 Future user needs	16
6.14.1 Software update	16
6.14.2 Start-up configuration	16
6.14.3 Configuration set management	16
6.14.4 Sequence of constructs within dynamic objects	16
7 High-level design	17
7.1 Authenticate users design overview	17
7.2 Control access to data design overview	17
7.3 Monitor failed access to the field device	18
7.4 Manage the field device design overview	18

7.5	Monitor field device components.....	18
7.6	Receive notification of triggers firing design overview	18
7.6.1	General	18
7.6.2	Triggers.....	19
7.6.3	Action selection	19
7.6.4	Notification factory.....	20
7.6.5	Notification channel.....	20
7.7	Manage device-specific notifications design overview	20
7.8	Log system events design overview.....	20
7.9	Log user-defined data snapshots design overview.....	20
7.10	Record a series of data snapshots design overview.....	21
7.11	Issue trigger-based commands design overview	22
7.12	Configure a complex device design overview.....	23
7.13	Efficient exchange of data design overview	23
7.14	Triggers.....	23
7.14.1	General.....	23
7.14.2	Scheduled triggers design overview	23
7.14.3	Day plan triggers design overview	24
7.14.4	Condition-based triggers design overview.....	25
8	Requirements.....	26
8.1	Action feature.....	26
8.1.1	Action feature definition.....	26
8.1.2	Action feature data exchange requirements.....	27
8.1.3	Action feature functional requirements.....	28
8.1.4	Action feature performance requirement.....	28
8.2	Clock feature.....	28
8.2.1	UTC Clock.....	28
8.2.2	Local clock.....	30
8.2.3	Daylight saving time.....	30
8.3	Command feature	31
8.3.1	Command feature definition	31
8.3.2	Command feature data exchange requirements.....	31
8.3.3	Command feature capability requirements.....	32
8.3.4	Command feature functional requirements.....	32
8.4	Conditional trigger feature.....	33
8.4.1	Conditional trigger feature definition.....	33
8.4.2	Conditional trigger feature data exchange requirements.....	33
8.4.3	Conditional trigger feature functional requirements.....	34
8.4.4	Conditional trigger feature capability requirements.....	34
8.5	Controller feature	39
8.5.1	Controller feature definition	39
8.5.2	Controller feature data exchange requirements.....	39
8.5.3	Controller feature capability requirements.....	41
8.5.4	Controller performance requirements.....	42
8.6	Day plan feature.....	42
8.6.1	Day plan feature definition.....	42
8.6.2	Day plan feature data exchange requirements.....	42
8.6.3	Day plan feature functional requirements.....	44
8.7	Dynamic object feature.....	44
8.7.1	Dynamic object feature definition.....	44
8.7.2	Dynamic object feature data exchange requirements.....	44
8.7.3	Dynamic object feature capability requirements.....	45
8.7.4	Dynamic object feature functional requirement.....	46
8.7.5	Dynamic object feature performance requirement.....	46
8.8	Field device feature	46
8.8.1	Field device definition	46
8.8.2	General field device features	47
8.8.3	Ambient air temperature.....	48

8.8.4	Ambient light	48
8.8.5	Ambient relative humidity	49
8.8.6	Auxiliary bi-directional port	49
8.8.7	Auxiliary input port	49
8.8.8	Auxiliary output port	49
8.8.9	Field device air conditioner	49
8.8.10	Field device battery	50
8.8.11	Field device dehumidifier	50
8.8.12	Field device doors	51
8.8.13	Field device fans	51
8.8.14	Field device generator	51
8.8.15	Field device heaters	52
8.8.16	Field device humidity	52
8.8.17	Field device mains power	52
8.8.18	Field device power supply	53
8.8.19	Field device processor temperature	53
8.8.20	Field device solar power	53
8.8.21	Field device temperature	54
8.8.22	Field device thermostat	54
8.8.23	Field device wind power	54
8.9	File feature	55
8.9.1	File feature definition	55
8.9.2	File feature data exchange requirements	55
8.10	Logging feature	55
8.10.1	Logging feature definition	55
8.10.2	Logging feature data exchange requirements	55
8.10.3	Logging feature functional requirements	57
8.10.4	Logging feature capability requirements	57
8.11	Notification feature	57
8.11.1	Notification channel	57
8.11.2	Notification factory	65
8.11.3	Independent notification	67
8.11.4	Notification aggregator	68
8.12	Owner feature	70
8.12.1	Owner feature definition	70
8.12.2	Owner feature data exchange requirements	70
8.13	Recording feature	71
8.13.1	Recording feature definition	71
8.13.2	Recording feature data exchange requirements	71
8.13.3	Recording feature capability requirements	73
8.13.4	Recording feature functional requirement	73
8.14	Scheduled trigger feature	73
8.14.1	Scheduled trigger feature definition	73
8.14.2	Scheduled trigger feature data exchange requirements	73
8.14.3	Scheduled trigger feature functional requirements	74
8.14.4	Scheduled trigger feature capability requirements	74
8.15	Secure communications feature	74
8.15.1	Secure communications feature definition	74
8.15.2	Secure communications protocol requirements	75
8.15.3	Secure communications conformance requirements	75
8.16	SNMP target feature	75
8.16.1	SNMP target feature definition	75
8.16.2	SNMP target feature data exchange requirements	75
8.16.3	SNMP target feature capability requirements	76
8.17	Start-up feature	76
8.17.1	Start-up feature definition	76
8.17.2	Start-up feature data exchange requirements	77
8.18	Supplemental roadside sensors and actuators (SRSA) feature	77
8.18.1	SRSA feature definition	77

8.18.2	SRSA feature data exchange requirements	77
8.18.3	SRSA feature capability requirements	78
8.18.4	SRSA feature performance requirements	79
8.18.5	SRSA feature supplemental requirements	79
8.19	System log feature	79
8.19.1	System log feature definition	79
8.19.2	System log feature data exchange requirements	80
8.19.3	System log feature capability requirements	80
8.20	Transaction feature	80
8.20.1	Transaction feature definition	80
8.20.2	Transaction feature data exchange requirements	81
8.20.3	Transaction feature functional requirements	81
8.21	View-based access control model (VACM) feature	81
8.21.1	VACM feature definition	81
8.21.2	VACM feature data exchange requirements	81
9	Dialogues	82
9.1	General dialogue rules	82
9.1.1	Management station initiated	82
9.1.2	SNMP agent performance requirements	82
9.1.3	Generic and custom dialogues	82
9.2	Generic dialogues	83
9.2.1	Get elemental data	83
9.2.2	Set elemental data	83
9.2.3	Walk data	83
9.2.4	Get bulk data	83
9.2.5	Get tabular data	83
9.2.6	Set tabular data	83
9.2.7	Get data column	84
9.2.8	Get counters	84
9.2.9	Get data from dynamic table entry	84
9.2.10	Get row status of dynamic table entry	84
9.2.11	Configure entry of a dynamic table	85
9.2.12	Configure entry of a dynamic table with TestAndIncr	86
9.2.13	Toggle active status of a dynamic table entry	86
9.2.14	Delete entry from a dynamic table	87
9.2.15	Send a notification	87
9.2.16	Retrieve dynamic object data	87
9.2.17	Set dynamic object data	87
9.2.18	Retrieve a file	88
9.2.19	Generate a file	88
10	Security	88
10.1	Vulnerabilities	88
10.2	Authentication and access control	88
10.3	Encryption	89
10.4	Security recommendations	89
	Annex A (informative) Conformance	90
	Annex B (informative) Management information base (MIB) summary	93
	Bibliography	95

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 204, *Intelligent transport systems*.

A list of all parts in the ISO 26048 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.

Intelligent transport systems — Field device Simple Network Management Protocol (SNMP) data interface —

Part 1: Global objects

1 Scope

Field devices are a key component in intelligent transport systems (ITS). Field devices include traffic signals, message signs, weather stations, traffic sensors, roadside equipment for connected ITS environments, etc.

The ISO 26048 series defines data that can be used to manage field devices, including device configuration, control and monitoring. Field devices can be quite complex, necessitating the standardization of many data concepts for exchange. As such, the ISO 26048 series is divided into several individual parts. This document (ISO/TS 26048-1) introduces the ISO 26048 series, provides content that is normatively referenced in subsequent parts, and defines data that is applicable to the management of a wide range of field devices.

The scope of the ISO 26048 series does not define the logic used by the management station, the underlying protocols used to exchange the defined data elements, or internal design of the field device. However, the ISO 26048 series does define functional requirements on the interface and assumes an interface based on an SNMPv3 environment as specified by ISO 15784-2.

NOTE Many of the concepts defined in this document were derived from NTCIP 1103^[1] and NTCIP 1201^[2], however, the design has been updated to better address security concerns. It is expected that future versions of NTCIP will migrate to the design defined in this document.

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 15784-2, *Intelligent transport systems — Data exchange involving roadside modules communication — Part 2: Centre to field device communications using Simple Network Management Protocol (SNMP)*

ISO/IEC 8825-1, *Information technology — ASN.1 encoding rules — Part 1: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*

ISO/IEC 8825-7, *Information technology — ASN.1 encoding rules — Part 7: Specification of Octet Encoding Rules (OER)*

ISO/IEC/IEEE 24765, *Systems and software engineering — Vocabulary*

ISO/TS 14812, *Intelligent transport systems — Vocabulary*

RFC 3415, *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*

RFC 5424, *The Syslog Protocol*

RFC 5676, *Definitions of Managed Objects for Mapping SYSLOG Messages to Simple Network Management Protocol (SNMP) Notifications*

RFC 8446, *The Transport Layer Security (TLS) Protocol Version 1.3*

FILE TRANSFER PROTOCOL S.S.H. <https://datatracker.ietf.org/doc/html/draft-ietf-secsh-filexfer-02>, October 2001.

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC/IEEE 24765, ISO/TS 14812 and the following 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 command generator

Simple Network Management Protocol (SNMP) application (3.19) that monitors and manipulates management information

3.2 command responder

Simple Network Management Protocol (SNMP) application (3.19) that provides access to management information

3.3 control object

writable *Simple Network Management Protocol (SNMP) object (3.23)* used to request immediate activation of a feature or state of the *field device (3.5)*

Note 1 to entry: Although control objects are writable, they are not considered part of the agent configuration and are therefore not *database parameters (3.4)*.

3.4 database parameter

writable Simple Network Management Protocol (SNMP) object used to configure an *SNMP agent (3.18)* and that can be set and validated using a single SNMP SetRequest-PDU

3.5 field device

fixed or portable roadside module that includes a *Simple Network Management Protocol (SNMP) agent (3.18)*

3.6 fire

start a process when a *trigger (3.28)* value transitions from false to true

3.7 inform

notification (3.11) sent with an expectation of an acknowledgement

3.8 interrelated parameter

writable *Simple Network Management Protocol (SNMP) object (3.23)* used to configure an *SNMP agent (3.18)* and that is designed to require the use of multiple SNMP SetRequest-PDUs prior to validating and implementing a new value

Note 1 to entry: The need for multiple SetRequest-PDUs is typically due to the number of other SNMP objects that are associated with the object. For example, a table can contain a column indicating a percent value where all rows of the table are supposed to equal 100. Configuring all rows within the table at once can exceed the size limitations on a single PDU. In this case, the designer can declare the SNMP object to be an interrelated parameter.

Note 2 to entry: Interrelated parameters are set using the transaction feature defined in [8.21](#).

3.9

log

registry of data *snapshots* (3.17) within a *Simple Network Management Protocol (SNMP) agent* (3.18) that can be retrieved by an *SNMP manager* (3.22)

3.10

management station

system that manages one or more *field devices* (3.5) with a *Simple Network Management Protocol (SNMP) manager* (3.22)

3.11

notification

Simple Network Management Protocol (SNMP) message from a *SNMP agent* (3.18) that is generated independently from any explicit request

Note 1 to entry: While a notification is not generated in response to any explicit request, it can be generated based on configured parameters stored within the *SNMP agent* (3.18).

3.12

notification originator

Simple Network Management Protocol (SNMP) application (3.19) that initiates asynchronous messages

3.13

notification receiver

Simple Network Management Protocol (SNMP) application (3.19) that processes asynchronous messages

3.14

principal

entity on whose behalf services are provided or processing takes place

EXAMPLE 1 An individual.

EXAMPLE 2 An application.

3.15

proxy forwarder

Simple Network Management Protocol (SNMP) application (3.19) that forwards messages between entities

Note 1 to entry: Proxy forwarder applications typically change the protocol or message model as a part of their functionality.

3.16

response time

time from the receipt of a Confirmed Class pduType by the *command responder* (3.2) to the sending of the response PDU by the command responder

Note 1 to entry: For this document, the response time is measured at the application programming interface of the command responder. Any delays imposed within the lower layers are network delays and are not included in the response time.

3.17

snapshot

data snapshot

information captured when a *trigger* (3.28) fires (3.6) within a *Simple Network Management Protocol (SNMP) agent* (3.18)

Note 1 to entry: A snapshot can be used in the generation of an SNMP *notification* (3.11) or the creation of a new entry within a *log* (3.9).

3.18

SNMP agent

Simple Network Management Protocol (SNMP) entity (3.20) containing one or more *command responder* (3.2) and/or *notification originator* (3.12) applications

3.19

SNMP application

application that provides specific functional processing of Simple Network Management Protocol (SNMP) management data

3.20

SNMP entity

implementation of one or more Simple Network Management Protocol (SNMP) message processing models with one or more associated *SNMP applications* ([3.19](#))

Note 1 to entry: An SNMP entity may also support one or more security models.

3.21

SNMP field manager

fixed or portable roadside module that includes a *Simple Network Management Protocol (SNMP) agent* ([3.18](#)) and an *SNMP manager* ([3.22](#))

Note 1 to entry: SNMP field managers are *field devices* ([3.5](#)) that can control other field devices.

3.22

SNMP manager

Simple Network Management Protocol (SNMP) entity ([3.20](#)) containing one or more *command generator* ([3.1](#)) and/or *notification receiver* ([3.13](#)) applications

3.23

SNMP object

instance of a data element within a *field device* ([3.5](#)) that can be exchanged via Simple Network Management Protocol (SNMP)

Note 1 to entry: SNMP objects can be categorized as *control objects* ([3.3](#)), *database parameters* ([3.4](#)), *interrelated parameters* ([3.8](#)), or *status objects* ([3.26](#)).

3.24

SNMP target

Simple Network Management Protocol (SNMP) entity ([3.20](#)) to which another SNMP entity can send requests or *notifications* ([3.11](#))

Note 1 to entry: *SNMP agents* ([3.18](#)) use the concept of a target to identify the *SNMP manager* ([3.22](#)) to which a notification is to be sent.

Note 2 to entry: *Field devices* ([3.5](#)) can be configured to request data from other field devices to use in their expression or *trigger* ([3.28](#)) logic.

Note 3 to entry: Field devices can be configured to control other field devices in response to a trigger firing.

3.25

standardized request

get or set request that conforms to the specification of a request message included in a dialogue defined within a document approved by a standards development organization

3.26

status object

read-only SNMP object

Note 1 to entry: Status objects are typically used to report the conditions that can be monitored by the *SNMP agent* ([3.18](#)).

3.27

trap

notification ([3.11](#)) sent without any expectation of an acknowledgement

3.28

trigger

condition that evaluates to a Boolean value

4 Symbols and abbreviated terms

ARC-IT	architecture reference for cooperative and intelligent transportation
ASCII	American standard code for information interchange
ASN.1	abstract syntax notation one
C-PICS	consolidated protocols implementation conformance statement
DTLS	datagram transport layer security
(D)TLS	datagram transport layer security or transport layer security
FTRT	feature-to-requirement traceability
IETF	Internet Engineering Task Force
IP	internet protocol
IPsec	internet protocol security
ISO	International Organization for Standardization
ITS	intelligent transport systems
ITS-S	ITS station
MIB	management information base
NTCIP	national transportation communications for ITS protocol
NTFT	need-to-feature traceability
OID	object identifier
PDU	protocol data unit
PT	predicate traceability
RFC	request for comments
RTM	requirements traceability matrix
SMI	structure and management of information
SNMP	simple network management protocol
SRSA	supplemental roadside sensor and actuator
TCP	transport control protocol
TLS	transport layer security
UDP	user datagram protocol
USM	user-based security model
VACM	view-based access model

5 Conventions and architecture

5.1 ISO maintenance portal

This document is accompanied by electronic attachments that form an integral part of this document and are available for download through the ISO maintenance portal at <https://standards.iso.org/iso/ts/26048/-1/ed-1/en>.

5.2 MIB files

5.2.1 MIB format

This document normatively incorporates several ASCII text files that define the `OBJECT-TYPES` and supporting information used to implement the requirements of this document. Each of these files conforms to RFC 2578, RFC 2579, and RFC 2580 and is known as a management information base (MIB).

The MIBs associated with this document enhance the `DESCRIPTION` clause of the `OBJECT-TYPE` macro, as defined in RFC 2578, by using the following textual subclauses, as needed:

- a) `<Definition>` — The normative definition of the described item. This subclause should be present for every `OBJECT-TYPE`, `OBJECT-IDENTITY`, `MODULE-IDENTITY`, and `TEXTUAL-CONVENTION` macro.
- b) `<Format>` — Explains how values are to be interpreted. This subclause should be used as needed.
- c) `<Parameter Type>` — This subclause identifies the types of data that the `OBJECT-TYPE` represents and the operations that are allowed under different scenarios. This information is used to unambiguously interpret some of the requirements of this document; it should be present for every `OBJECT-TYPE` that has a `MAX-ACCESS` of `read-only`, `read-write`, or `read-create`. For example, the value within this subclause affects the calculation of the configuration identifier (8.5.2.14) and the response functionality of the SNMP interface when using the transaction feature (8.21). The valid values for parameter type are as follows:
 - 1) `Status` — Read-only object-type that reports a condition that can be monitored.
 - 2) `Control` — Writeable object-type used to request real-time activation of a feature of a device. In some cases, a control object can also be used to report status.
 - 3) `Configuration` — Writeable object-type used to configure the SNMP agent where the parameter can be set and validated using a single SNMP set operation. All configuration parameters shall be stored in non-volatile memory.
 - 4) `Interrelated configuration` — Writeable object-type used to configure the SNMP agent where the parameter has sufficient interrelationships with other object-types to typically require multiple SNMP set operations or a complex validation check that can consume more time than is reasonable for a traditional SNMP response. When a device includes interrelated parameter object types, it defines a mechanism by which the object types can be safely configured. One such approach is to use the transaction feature (8.21).
- d) `<Informative>` — Provides any non-normative, supplemental information about the object-type. For example, this can be used to provide historic information about how the definition has evolved over time. This subclause should be used as needed.
- e) `<Superseded by>`, `<Supplanted by>`, or `<Withdrawn>` — When the `STATUS` of the `OBJECT-TYPE` is “deprecated” or “obsolete”, one of these three subclauses should be used to further indicate the status of the underlying feature as follows:
 - 1) The “superseded by” option indicates that the object has been replaced with a new object that interacts with the same functionality of the device; in other words, the values of the old and new objects are interrelated. For example, an object that measures distance in metres can be superseded by an object that measures distance in millimetres. In an implementation that supports both the old

and new objects, the old object would always have the same value as the new object divided by 1000 (rounded or truncated to the nearest integer). The `<Superseded by>` heading should be followed with the name of the replacement object.

- 2) The “supplanted by” option indicates that the object has been replaced with a new object that references separate, but similar, functionality within the device. Sometimes, changes to a MIB require a fundamental redesign such that the old objects cannot be used to manage the new design. In this case, the `<Supplanted by>` heading should be followed by the name of the near-equivalent replacement object.
 - 3) The “withdrawn” option indicates that the object has been withdrawn without any equivalent replacement. In this case, there typically is not any text after the `<Withdrawn>` heading.
- f) `<Object Identifier>` — An informative rendering of the full object identifier of the `OBJECT-TYPE` in dotted notation. The object identifier is normatively assigned per the rules defined in RFC 2578; however, this field provides a consolidated representation of the assignment to assist human readers. This subclause should be present for every `OBJECT-TYPE`, `OBJECT-IDENTITY`, and `MODULE-IDENTITY` macro.

The subclauses can appear in any order, although the above order is preferred. A document that imports entities from MIBs that do not include parameter type information should indicate how the intended parameter type for each imported `OBJECT-TYPE`.

5.2.2 MIB access

The MIB files associated with this document can be accessed electronically from the ISO maintenance portal as indicated in [5.1](#).

5.2.3 MIB filenames

The filename for each component MIB is the name of the MIB with a ".mib" extension. A complete listing of MIBs associated with this document is contained in [Annex B](#).

5.3 ASN.1

This document contains references to and explanations of ASN.1 data concepts^[3] within its text. In all cases, the ASN.1 terms are presented in a fixed width font (e.g. `such as this`) to distinguish these terms from normal English.

5.4 Conformance

Conformance and consistency to this document is defined by the traceability tables available through the ISO Maintenance Portal, including the need-to-feature traceability (NTFT) table, the feature-to-requirements traceability (FTRT) table, and the requirements traceability matrix (RTM).

Each item of conformance in this document is worded as if it has been selected in these tables. For example, requirements are stated as simple "shall" statements even if they are listed as optional in the FTRT table. The applicability of each item of conformance is defined in the traceability tables as discussed further in [Annex A](#).

5.5 SNMP terminology

Terminology between the different versions of SNMP is slightly different. For the purposes of the ISO 26048 series, the terminology of SNMPv3 is adopted.

5.6 Architecture

5.6.1 ITS services

The ISO 26048 series defines mechanisms by which field devices can be monitored, configured and controlled. Field devices may be used to support almost any ITS service with a roadside component. Reference [4] provides a listing of known ITS services.

5.6.2 Functional view of interface

The ISO 26048 series is concerned with defining the data concepts used to manage a field device. The data concepts defined in this document have been defined based on needs derived from an analysis of various services contained within the architecture reference for cooperative and intelligent transportation (ARC-IT)[5].

5.6.3 Physical view

[Figure 1](#) depicts the physical view of this interface using the graphical conventions defined by ARC-IT and documented in ISO 14813-5:2020, Annex B.

The management station of the field device is shown in grey, indicating that it can be any type of physical object, such as a central system, another field device, a maintenance laptop or any other device that supports the defined interface.

The field device is shown in orange, indicating that it is in the field (e.g. along the roadside). It shall have a connection to the management station and may have any number of connections to other ITS stations or external systems.

The figure indicates two high-level information transfers between these physical objects. The first is the "configuration and commands" information flow from the management station to the field device. The second is the "status and notifications" information flow from the field device to the management station. Both flows are shown in green indicating that authentication is required, and both are shown with a single arrowhead indicating a unicast transfer.

NOTE 1 This document is based on the use of SNMP, which implements a GET/SET paradigm where there is a command generator (i.e. the management station) and a command responder (i.e. the field device). In SNMP, both flows are likely to be initiated by the management station; the direction of the arrows in [Figure 1](#) indicate the primary direction of information flow.

NOTE 2 A single SNMP entity can act as both a command generator (e.g. sending requests to other field devices) and as a command responder (e.g. responding to requests from a centre or other field device) simultaneously. The term "SNMP field manager" is used to specify this type of dual-function device.

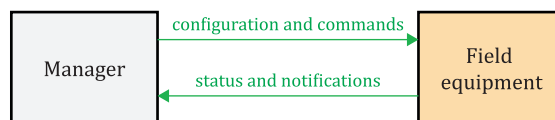


Figure 1 — Physical view of interface

5.6.4 Communications view

[Figure 2](#) depicts how the ISO 26048 series is intended to relate to other standards using the ITS Station (ITS-S) architecture, as defined in ISO 21217.

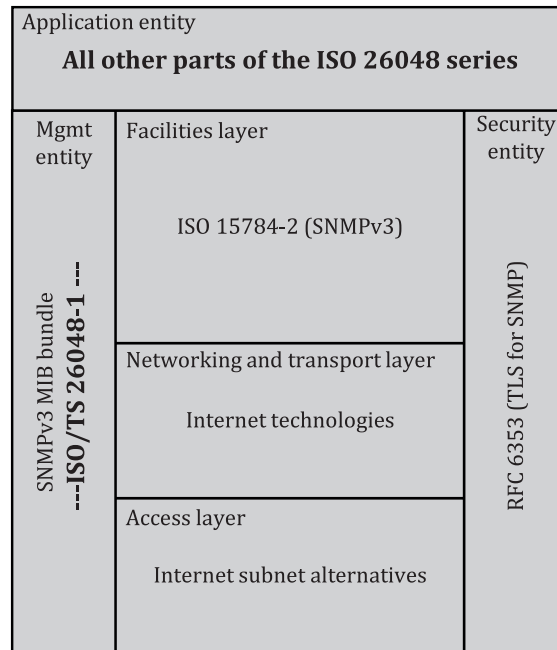


Figure 2 — Typical communications stack

This document (ISO 26048-1) defines features that are a part of the management entity of the ITS-S architecture. This includes management information that indirectly supports device-specific functionality. The management layer typically also supports other data to manage the communications stack as defined by a variety of SNMPv3 MIBs (e.g. the management information referenced by ISO 15784-2).

EXAMPLE 1 [8.2.1](#) defines the UTC clock feature, which can be used to support end-device functionality, such as selecting a time-of-day action plan.

Other parts of the ISO 26048 series address device-specific functionality and are considered part of the definition of ITS-S application entity.

EXAMPLE 2 ISO 26048-3:—¹⁾ is intended to define management information related to controlling and monitoring a variable message sign.

The information defined for the ITS-S management entity and ITS-S application entity can theoretically be exchanged over any communications stack, but is designed to be exchanged using SNMPv3, as described in ISO 15784-2. This requires support for the transport layer security (TLS) transport model, as defined in References [8] and [9]. These standards are typically exchanged using well-known internet protocols, such as UDP/IP or TCP/IP over any number of access layers.

5.7 Constraints

5.7.1 Table design constraint

Tables that are likely to contain different definitions for different management stations are designed to easily restrict access to authorized management stations and avoid inadvertent conflicts.

EXAMPLE A log can be designed to support a scheme where an administrative management station that has universal rights does not inadvertently change the definition of a log for another management station.

1) Under development. Stage at the time of publication: ISO/AWI TS 26048-3:2024.

5.7.2 Access constraints

Under all circumstances, the field device only allows each management station to access data to which it is explicitly authorized.

NOTE This includes data that is recorded in logs, sent in notifications, and other indirect access mechanisms.

5.8 Outline

This document presents a separate section for each step in the systems engineering process.

- a) User needs — Each user need describes a capability that a management station can potentially need to perform. User needs are written from the perspective of a management station and include a justification for the need.
- b) High-level design — Each user need is analysed and decomposed into a set of features. Features can potentially support multiple user needs. The applicability of each feature within a user need is defined by the NTFT table, which is available on the ISO Maintenance Portal as discussed in [5.1](#).
- c) Requirements — Each feature is decomposed into a set of formal requirements, including data exchange requirements and other requirements (e.g. performance, constraints). The applicability of each requirement within a feature is defined by the FTRT table, which is available on the ISO Maintenance Portal as discussed in [5.1](#).
- d) Dialogues — Each data exchange requirement is implemented using exactly one standardized dialogue coupled with a defined set of objects. The standardized dialogue is defined as a sequence of SNMP operations. The traceability between data exchange requirements and their associated dialogues is defined in the RTM, which is available on the ISO Maintenance Portal as discussed in [5.1](#).
- e) Management Information Base — Each data exchange requirement is satisfied using one or more managed objects to be included in the associated dialogue. Objects are instances of object-types, which are defined in electronic MIB(s). The traceability between each data exchange requirement and the associated object-types are defined in the RTM. The RTM and MIBs defined by this document are available on the ISO Maintenance Portal as discussed in [5.1](#). Other MIBs (e.g. those in RFCs) are available in the referenced document.

5.9 Requirements pattern

Features are defined as a package of requirements that follow a logical pattern.

To implement the pattern, each feature is divided into one or more elemental features that correspond to a high-level design element (e.g. a table). For example, a feature can include one table designed to manage the collection of information and a separate table to store collected information. In this case, the parent feature is divided into two sub-features: one for each table.

The requirements for each elemental feature are developed according to a pattern of specific types of requirements that has emerged through repeated analysis of multiple field devices. Each elemental feature is analysed to determine if each specific type of requirement is applicable, and if so, it is included in the package of requirements. Data exchange requirements follow a rather precise pattern whereas other requirements are less well defined. While this pattern-based approach does not necessarily address every possible requirement, it addresses the bulk of requirements in a manner that allows a high level of confidence that all requirements have been addressed.

The types of requirements in the pattern are listed below. The terminology used for these requirements has been chosen to allow the requirements to be listed in mostly an alphabetical order while also providing an order that is semantically logical.

- a) Data exchange requirements: Data exchange requirements are divided into the following subtypes:
 - 1) Change status of an item of an elemental feature: This requirement allows the management station to enable or disable a singular item (e.g. row) of an elemental feature (e.g. table) while retaining

its configuration. This is typically implemented with the "active" and "notInService" values of the `RowStatus` textual convention, as defined in RFC 2579.

- 2) Configure an item of an elemental feature: This requirement allows the management station to change the configuration parameters that controls how a singular item of an elemental feature operates.
 - 3) Confirm configuration of an item of an elemental feature: This requirement allows a management station to retrieve the current configuration and is the read operation that corresponds to the write operation of the configure requirement.
 - 4) Control the operation of a feature: This requirement allows a management station to set the operational mode of a feature (e.g. selecting among multiple allowed algorithms, starting a process, aborting a process).
 - 5) Clear an elemental feature: This requirement allows a management station to permanently delete all items within an elemental feature (e.g. all rows of a table).
 - 6) Delete an item of an elemental feature: This requirement allows a management station to permanently delete a single item of an elemental feature (e.g. a single row of a table).
 - 7) Determine capabilities of an elemental feature or a feature: This requirement allows a management station to discover the capabilities of a feature within the specific implementation. When features include optional capabilities, management stations need to be able to discover which optional capabilities are supported by a specific implementation.
 - 8) Monitor statistics related to an elemental feature or feature: This requirement allows a management station to learn about the past operation of the feature by discovering how often events of interest have occurred. The data for each event of interest is typically implemented with some sort of counter textual convention.
 - 9) Monitor status of an elemental feature: This requirement allows a management station to determine the operational or other status of a feature (e.g. all items are disabled in some way, an error has occurred that affects all items of the feature). The status data is typically satisfied by an enumerated data element.
 - 10) Obtain results of an elemental feature: Features can perform some sort of functionality that processes information to produce new data. This requirement allows a management station to retrieve the data generated by the feature.
 - 11) Retrieve inputs for an elemental feature: Features can rely on inputs, which can be from sensors of the management station setting input values. This requirement allows a management station to retrieve these input values (e.g. for further processing, to better understand the operation of the field device).
 - 12) Set inputs for an elemental feature: Features can rely on inputs received from the management station. Inputs are distinct from configuration parameters in that they are not intended to be long-term values.
- b) Capability requirements: Capability requirements identify specific features to be supported by a field device. Many capabilities are implicitly defined within aspects of other requirements; however, when capabilities can vary by implementation, it is important to document the capabilities explicitly so that actors can identify precisely what is required of or supported by an implementation. Capability requirements typically identify whether specific capabilities are supported (e.g. whether a specific algorithm is supported) or identify sizing limitations of a capability (e.g. the number of items of an elemental feature).
- c) Design constraints: Design constraints identify additional constraints that are to be considered during the design of the feature. For example, they can require the use of particular data standards or identify an operational environment that requires an additional index to a table that would potentially not otherwise be considered.

- d) **Functional requirements:** These requirements identify the actions to be performed by the device in association with the feature. In many cases the actions are implicitly defined by the feature but in some cases, there can be a need to explicitly define the details.
- e) **Information requirements:** Most information requirements are fully contained within the MIB files, but in some cases, it can be useful to specify complex data structures that are used by the design.
- f) **Performance requirements:** These requirements address latency, timing, resolution, precision and other concerns that deal with the quality of the data exchanged.

It is recognized that some requirements can potentially be classified under more than one of the above requirement types. The intent of the pattern is to aid the designer in considering different aspects of each feature to ensure that all requirements are captured and to aid the user of the document by providing a logical and consistent ordering of requirements.

6 User needs

6.1 Authenticate users

A management station needs to ensure that the principal (i.e. user) submitting each request is authenticated to be the claimed principal.

6.2 Control access to data

A management station needs to ensure that users are only allowed to access data according to the authority that they have been granted.

6.3 Monitor failed access to the field device

A management station needs to be able to monitor failed attempts to access the field device so that unauthorized access attempts can be monitored and corrective actions can be taken.

6.4 Manage the field device

A management station needs to be able to control, identify and monitor the overall capabilities and health of the field device controller and its cabinet to discover anomalous conditions that can affect its operation or security. This will assist the management station in confirming which controller(s) are in a cabinet, as well as the type and specific instance of each controller and the high-level capabilities offered by the device as well as performing proper maintenance actions.

EXAMPLE A management station that is receiving unexpected errors can verify which device it is communicating with as a part of a debugging process and can determine if the device configuration has been changed since its last known state so that the appropriate action can be taken if access has not been authorized.

6.5 Monitor field device components

6.5.1 Monitor ambient environment

6.5.1.1 Monitor ambient air temperature

A management station needs to be able to monitor the ambient air temperature.

6.5.1.2 Monitor ambient relative humidity

A management station needs to be able to monitor the ambient relative humidity.

6.5.1.3 Monitor ambient light

A management station needs to be able to monitor the current ambient light situation.

6.5.2 Monitor field device enclosure

6.5.2.1 Monitor field device doors

A management station needs to be able to monitor the open/close status of each door associated with an enclosure of the field device to determine when equipment is being physically accessed.

6.5.2.2 Monitor field device enclosure air temperature

A management station needs to be able to monitor the air temperature within field device enclosures to ensure that components do not exceed their operating temperature ranges.

6.5.2.3 Monitor field device processor temperature

A management station needs to be able to monitor the processor temperature to ensure that it does not exceed its operating temperature range.

6.5.2.4 Monitor field device enclosure relative humidity

A management station needs to be able to monitor the relative humidity of the air within field device enclosures to ensure that components do not exceed their operating ranges.

6.5.3 Manage enclosure climate control components

6.5.3.1 Manage field device fans

A management station needs to be able to monitor and control the on/off status of each fan used to manage the temperature within field device enclosures.

6.5.3.2 Manage field device heaters

A management station needs to be able to monitor and control the on/off status of each heater used to manage the temperature within field device enclosures.

6.5.3.3 Manage field device dehumidifiers

A management station needs to be able to monitor and control the on/off status of each dehumidifier used to manage the humidity within field device enclosures.

6.5.3.4 Manage field device air conditioners

A management station needs to be able to monitor and control the on/off status of each air conditioner used to manage the temperature within field device enclosures.

6.5.3.5 Manage field device thermostat

A management station needs to be able to monitor and control the on/off status of each thermostat used to manage the temperature within field device enclosures.

6.5.4 Monitor field device power

6.5.4.1 Monitor field device power supplies

A management station needs to be able to monitor the status of each power supply that converts incoming power into the form required by components of the field device.

6.5.4.2 Monitor field device mains power

A management station needs to be able to monitor the status of the incoming mains power line, which is typically provided by the power grid to detect when this power is lost or becomes unstable.

6.5.4.3 Monitor field device battery power

A management station needs to be able to monitor the status of the battery power system to determine the quality of power and amount of charge available.

6.5.4.4 Monitor field device generator power

A management station needs to be able to monitor the status of the generator power to determine the quality of power being produced and the fuel reserve.

6.5.4.5 Monitor field device solar power

A management station needs to be able to monitor the status of the solar power system to determine the amount of power being generated.

6.5.4.6 Monitor field device wind power

A management station needs to be able to monitor the status of the wind power system to determine the amount of power being generated.

6.5.5 Manage auxiliary ports

6.5.5.1 Manage field device auxiliary bi-directional ports

A management station needs to be able to manage any bi-directional data ports offered by the field device for general-purpose operations (i.e. operations not defined by this document).

6.5.5.2 Monitor field device auxiliary inputs

A management station needs to be able to manage any input data ports offered by the field device for general-purpose input (i.e. inputs not defined by this document).

6.5.5.3 Manage field device auxiliary outputs

A management station needs to be able to manage any output data ports offered by the field device for general-purpose outputs (i.e. outputs not defined by this document).

6.6 Receive notification of triggers firing

When user-defined triggers fire, one or more independent management stations need to receive real-time notifications containing user-defined information. This will allow a management station to immediately become aware of information that can affect its operation or security without burdening the communications

channel with frequent polling for data that seldom changes. Multiple triggers can be monitored with different systems being notified based on the type of condition.

EXAMPLE 1 A management station wants to be immediately notified when the cabinet door opens so that an appropriate response can be initiated if the access is unauthorized.

EXAMPLE 2 A management station wants to have the maintenance system notified when the cabinet door opens and have the traffic management station notified when a new message is displayed on a sign.

6.7 Manage device-specific notifications

A management station needs to be able to manage the transmission of device-specific notifications so that they do not overwhelm the communications system.

6.8 Log system events

A management station needs to be able to retrieve errors, warnings and notices that the field device and any installed ITS applications have recorded. These messages are often created by developers to assist in debugging operations and include timestamps and severity levels. This need includes the ability to configure which messages should be maintained in the log and which should be reported.

6.9 Log user-defined data snapshots

One or more independent management stations need to be able to configure a field device to log data snapshots for later retrieval. This user need allows a management station to configure conditions that will automatically trigger the logging of snapshot data along with a timestamp. A management station can configure multiple triggers and multiple management stations can configure the same or different conditions.

EXAMPLE 1 A management station wants the device to record the number of vehicles counted every 15 min so that the management station can retrieve the information at the end of each day.

EXAMPLE 2 A management station wants to retrieve diagnostic-related snapshots (e.g. cabinet door open) separately from operation-related snapshots (e.g. a new message on a sign). This can be due to internal logic, or it can be because two separate management stations communicate with the device.

6.10 Record a series of data snapshots

One or more independent management stations need to be able to configure a field device to log a series of data snapshots based on a specific trigger being fired once. This user need allows a management station to configure conditions that will automatically trigger the logging of a series of data snapshots along with timestamps. A management station can configure multiple triggers and multiple management stations can configure the same or different conditions. Once the data snapshots have been logged, the management station can retrieve the information.

EXAMPLE A management station wants the device to record the count of the number of vehicles approaching an intersection each second after the signal changes to yellow for a period of 30 seconds to produce a better estimate of the delays at the signal.

6.11 Issue trigger-based commands

One or more independent management stations need to be able to configure an SNMP field manager to send a command to an SNMP agent when user-defined conditions are detected by the SNMP field manager. This will allow the field manager to alter the state of other field devices as triggers fire without burdening the communications channel between the centralized management station and the remote field location.

NOTE The SNMP field manager and SNMP agent can be in the same field device.

EXAMPLE A management station (e.g., a central system) wants to configure a field manager to automatically display a message on Device A whenever ice is detected on the roadway by Device B, where the field manager, Device A, and Device B are all located near each other and can communicate with minimal delays. The field manager can be in a separate physical unit or contained within either Device A or Device B.

6.12 Configure a complex device

For devices that have extensive inter-relationships among configuration parameters, a management station needs to provide all necessary configuration updates prior to the new configuration being validated and implemented.

EXAMPLE A phased-based traffic signal controller has many objects that are safety critical. When reconfiguring a traffic signal controller, it is important to ensure that the values are internally consistent to prevent any potential for unsafe operation. The number of inter-related parameters typically exceed what can be delivered in a single SNMP message. As a result, a distinct mechanism is required to buffer the initial set of changes and to perform validation once all parameters have been set.

6.13 Efficient exchange of data

One or more management stations need to repeatedly perform requests related to the same group of managed object values. Normal SNMP operations require the complete^[10] object identifier (OID) of every object value to be specified in each request. These OIDs are often 10-20 octets in length, which can significantly increase communications overhead when the values being retrieved are small (e.g. one-octet INTEGERS). This document defines a mechanism to allow a management station to configure dynamic objects, which can then be used to interact with multiple managed objects via a single OID.

EXAMPLE A management station polls a specific group of integer-based objects periodically to monitor the operation of a field device. The communication network used to connect to the field device includes a link that charges for data usage. Rather than requesting each status object individually, the management station wishes to configure a structure that can be repeatedly requested using less overhead.

6.14 Future user needs

6.14.1 Software update

Management stations need to update software and firmware within controllers. This user need will potentially be addressed in a future edition of this document.

6.14.2 Start-up configuration

Management stations need to configure the configuration files and parameters to use at start-up. This user need will potentially be addressed in a future edition of this document.

6.14.3 Configuration set management

Management stations need to be able to easily activate any of several previously downloaded configurations based on trigger conditions (e.g., based on time-of-day and day-of-week). This user need will potentially be addressed in a future edition of this document.

6.14.4 Sequence of constructs within dynamic objects

Management stations need to be able to easily define and use dynamic objects that contain the values in specified columns from specified rows of a table without specifying each object instance in order. This user need will potentially be addressed in a future edition of this document.

EXAMPLE When uploading or downloading the configuration of a field device, a management station needs to be able to retrieve all configuration parameters for each row within a table. If the total number of configuration parameters in the table is the product of the number of configuration parameters within a row multiplied by the number of rows. By comparison, this user need would allow a specification by defining the configuration parameters within a row plus the identifiers of the start and end row.

7 High-level design

7.1 Authenticate users design overview

User authentication is handled by the underlying protocol. It is envisioned that the underlying protocol will be ISO 15784-2, in which case, the authentication is provided by (D)TLS coupled with the TLS transport model and transport security model. If a different protocol is supported, it shall define how users are adequately authenticated.

7.2 Control access to data design overview

Data access is controlled using the view-based access control model (VACM), as defined in RFC 3415, coupled with a design of the OID tree^[10] to ensure that administrators can easily restrict access to branches of the tree based on intended access rights.

Configuring the VACM consists of the following major steps:

- a) discovering available contexts within the field device;
- b) configuring MIB views;
- c) assigning each user to a VACM group; and
- d) configuring the accessible MIB view for each VACM group based on the context and operation (e.g. read versus write) contained in the request.

A context represents a complete instantiation of management information. A field device can contain one or more contexts. For example, a single field device can be used to control the display for two signs (e.g. one facing north and one facing south). Such a device can define two contexts, one for each sign. Each context would contain a complete instance of the MIB with unique object instances for each. When appropriate, object instance values can be linked. For example, each context would likely have a different value for the currently displayed message while both contexts should have a common interpretation of the time of day. Contexts are defined by the device and cannot be configured by a management station.

NOTE Many field devices contain a single context and use an empty string to identify the context.

An MIB view represents a subset of objects available within a device to which a user should have access. An MIB view is configured by identifying portions of the OID tree to which access should be granted or prohibited.

EXAMPLE 1 A branch of the OID tree that is only intended to be accessible to administrators can be excluded from an MIB view intended to be used by technicians.

A VACM group represents zero or more authenticated users, where each user is identified by a security name and security model pair. When requests are received, the SNMP engine determines the security name and security model and uses the information to determine which group's access rights to use. The process used to authenticate this information is defined separately by the security model.

EXAMPLE 2 All field technicians using a particular security model can be assigned to the same group.

Once the necessary contexts, VACM groups, and MIB views are defined, access can be configured for each VACM group. The configuration includes separate assignments based on the type of request (e.g. get, set) as well as the security model and level (e.g. whether encryption is used).

EXAMPLE 3 The same user can be granted different access rights for a context related to a north facing sign than for a context related to a south facing sign.

EXAMPLE 4 The same user can be assigned different MIB views for get requests than for set requests.

EXAMPLE 5 The same user can be granted different access using encryption than when not using encryption.

7.3 Monitor failed access to the field device

Monitoring unauthorized access to the field device is achieved using permanently configured entries within the conditional trigger feature, as defined in [8.4](#), to monitor access attempts that use invalid credentials. When these triggers fire, they can be used to create a notification as discussed in [7.6](#) or generate a log entry as discussed in [7.9](#).

7.4 Manage the field device design overview

Monitoring the field device involves the field device feature, the start-up feature, and the cabinet feature. The object-types included within these features primarily consist of information that is either read-only or writable with no or minimal interactions with other management information from a semantic perspective.

7.5 Monitor field device components

Monitoring field device components is achieved by coupling the supplemental roadside sensors and actuators feature with the feature to be monitored.

7.6 Receive notification of triggers firing design overview

7.6.1 General

Notifications allow a field device to transmit unsolicited messages to management stations based on trigger conditions monitored by the field device. This document defines a way for users to configure the triggers, the data to be included in the resulting notifications, and the rules for managing notifications. These operations are achieved using a variety of features as depicted in [Figure 3](#), each of which needs to be configured for the feature to work as intended.

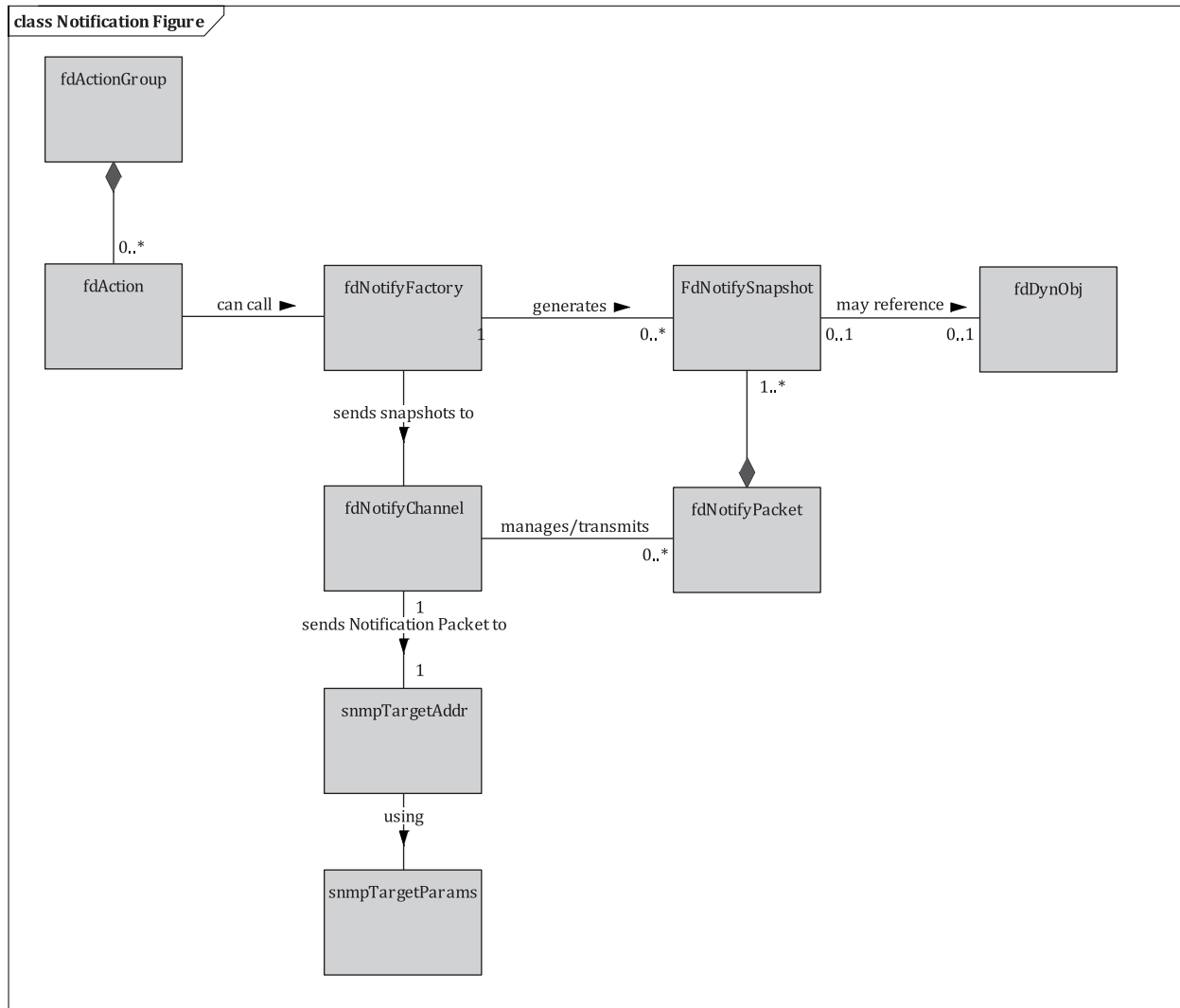


Figure 3 — Conceptual overview of notifications

7.6.2 Triggers

Although not shown in [Figure 3](#), the notification process starts when a trigger ([7.14](#)) fires. This document defines three trigger features that can be used for this purpose: the conditional trigger feature ([7.14.4](#)), the day plan feature ([7.14.3](#)) and the scheduled trigger feature ([7.14.2](#)). Any of these triggers can be used to call an action group ([8.1](#)).

In addition, SNMPv3 allows MIBs to define their own notifications. In this case, the notification definition explicitly defines the conditions under which the notification is sent. Within this document, these notifications are called "independent notifications". This document requires independent notifications to be managed by the same notification channel mechanism defined in [7.6.5](#).

7.6.3 Action selection

An action group can contain multiple actions, such as logging information within the field device, issuing a command to another field device, or generating a notification using the notification factory. This description concerns only the call to a notification factory.

7.6.4 Notification factory

When called, the notification factory captures the current value of a specified object and stores this value in a notification snapshot along with a timestamp and an identifier of the notification factory that created the snapshot. The object value recorded may be an `fdDynObjCurrentValue` (ISO26048-1-DynObj.mib), which can consolidate the values of multiple subordinate objects into a single SNMP object. The notification factory then passes the notification snapshot to the notification channel along with information about how the notification snapshot is to be handled (e.g. aggregated or not). Multiple notification factories may use the same notification channel.

7.6.5 Notification channel

If the notification snapshot is flagged for aggregation, the notification channel places the snapshot into one of its two buffered aggregate notification packets (i.e. either acknowledged or unacknowledged per the snapshot factory configuration); otherwise the notification channel assigns the notification snapshot to a non-aggregated notification packet (either acknowledged or unacknowledged according to the snapshot factory configuration).

Aggregated snapshots reduce overall communications overhead by combining multiple notification snapshots into a single packet but delay the delivery of snapshots. Aggregation is considered complete when either the aggregation time allowed for any contained snapshot is exceeded or when the maximum size of the aggregation is reached. Once the aggregation is complete, it is prepared for transmission and the next aggregated snapshot begins a new buffer.

Once a notification packet is ready to send, the notification channel sends the notification packet to the SNMP target while ensuring that the anti-streaming rate is not exceeded, which allows temporary queueing of notifications so that they do not overwhelm the communications channel.

The notification channel also manages the guaranteed delivery mechanism. While SNMP inform packets allow for confirmation, they do not provide any attempt to guarantee delivery (i.e. if acknowledgement is not received, the field device continues as it would otherwise). The guaranteed delivery mechanism defined by this document provides a much higher probability that the information will reach its target.

7.7 Manage device-specific notifications design overview

All device-specific notifications are configured to indicate the notification channel(s) to be used and whether they are queueable, require guaranteed delivery, or both. Each notification channel is then responsible for ensuring that each device-specific notification is managed using the same buffer and queue logic as with the trigger-initiated notifications ([7.6.5](#)).

7.8 Log system events design overview

Logging system events for debugging purposes is based on the syslog design in accordance with RFC 5424 and RFC 5676, as specified in the system log feature ([8.19](#)). This base design can be extended to support filtering of log entries through the system log filter feature, defined in [8.20](#). Both mechanisms reference the file feature ([8.9](#)) for exchanging log files.

7.9 Log user-defined data snapshots design overview

Logging snapshots is achieved using a variety of features as depicted in [Figure 4](#).

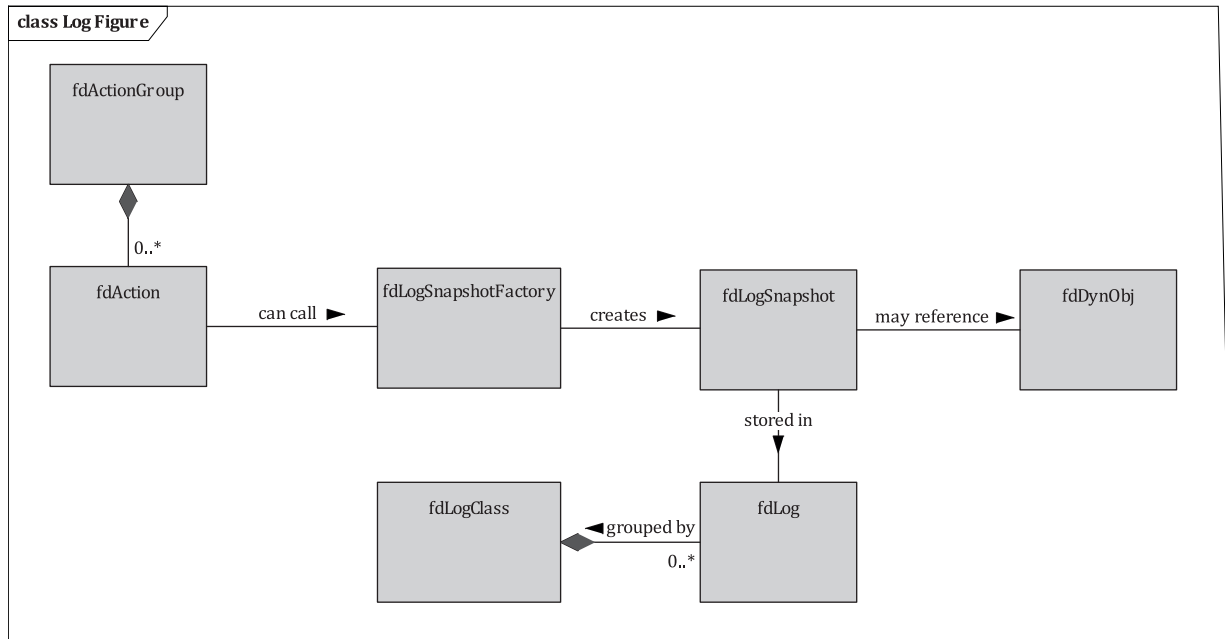


Figure 4 — Conceptual overview of logging

When a trigger (7.14) fires, it calls an action, which may direct the call to a log snapshot factory. When called, the log snapshot factory captures a defined object value from the device and records this in a new snapshot entry in the log class and associates it with a log class based on configured parameters.

While the snapshot only records a single object instance value, the object instance can be an instance of `fdDynObjCurrentValue`, which can contain multiple object instance values packaged in an efficient manner.

The log can be managed (e.g. fully or partially cleared) and monitored with statistics.

7.10 Record a series of data snapshots design overview

Recording a series of snapshots is achieved using a variety of features as depicted in Figure 5.

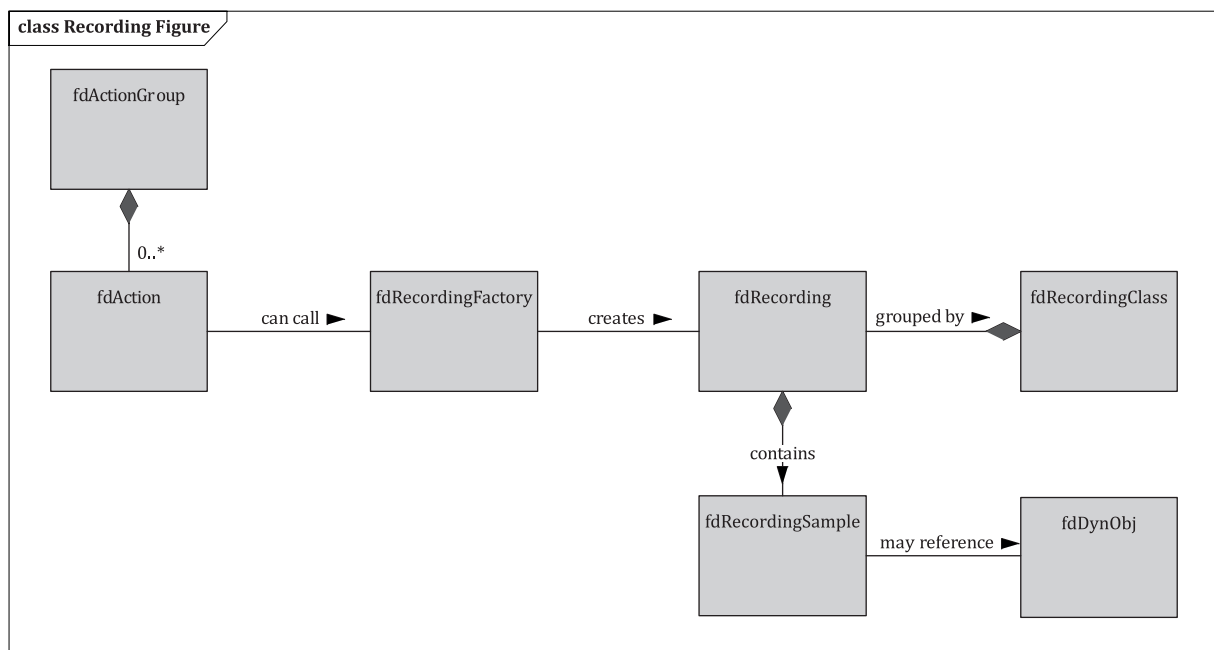


Figure 5 — Conceptual overview of recordings

When a trigger (7.14) fires, it calls an action, which may direct the call to a recording factory. When called, the recording factory initiates a new recording, associating it with a recording class according to the factory's configuration. If the recording is configured to include pre-event samples, the recording factory moves the configured number of previously recorded samples into the newly initiated recording. The recording factory continues capturing samples according to the configuration and stores each within the recording.

While each sample only records a single object instance value, the object instance can be an instance of `fdDynObjCurrentValue`, which can contain multiple object instance values packaged in an efficient manner.

The recording can be managed (e.g. fully or partially cleared) and monitored with statistics.

7.11 Issue trigger-based commands design overview

Issuing trigger-based commands is achieved using a variety of features as depicted in Figure 6.

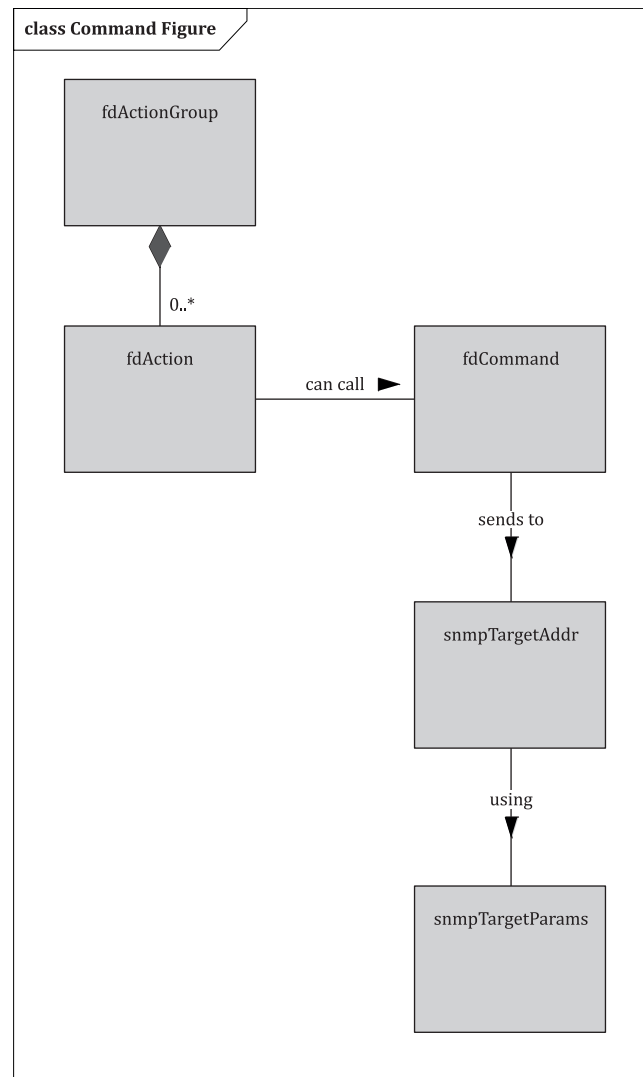


Figure 6 — Conceptual overview of commands

When a trigger (7.14) fires, it calls an action, which may direct the call to a command. When called, the SNMP field manager sends the configured SNMP set request to the identified SNMP target using the configured SNMP target parameters.