INTERNATIONAL STANDARD

ISO 24102-4

Second edition
2018-11

Intelligent transport systems — ITS station management —

Part 4: Station-internal management communications

Systèmes intelligents de transport — Gestion des stations ITS —
Partie 4: Communications de gestion interne à la station

Reference number
ISO 24102-4:2018(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

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 documents 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).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

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*.

This second edition cancels and replaces the first edition (ISO 24102-4:2013) which has been technically revised. It also incorporates the Amendment ISO 24102-4:2013/Amd1:2017.

A list of all parts in the ISO 24102 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.

NOTE The former ISO 24102-5 has been converted into a separate standard ISO 22418[1], as it is not a station management standard.

Introduction

This document is part of a series of International Standards for communications in intelligent transport systems (ITS) based on the ITS station and communications architecture specified in ISO 21217 and illustrated in Figure 1.

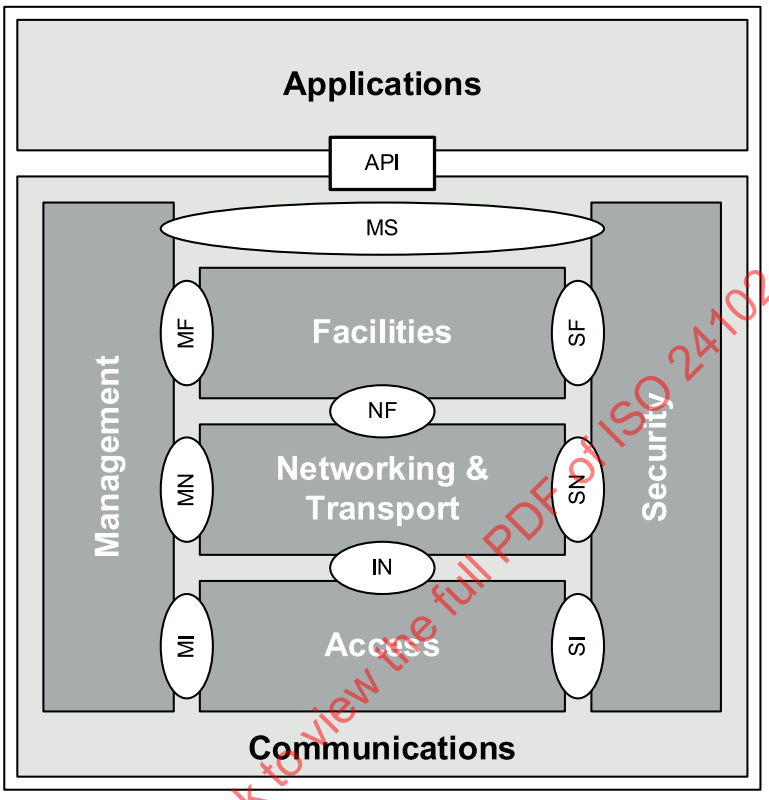

Figure 1 — ITS station reference architecture with named interfaces

This document is Part 4 of a multi-part standard which determines the intelligent transport systems (ITS) station-internal management communications that is architecturally located in the ITS station Management entity.

The ITS station management entity provides functionality related to the management of communication protocol layers (Access, Networking & Transport, Facilities), the Security entity, and the ITS Applications entity introduced in ISO 21217:2014 and presented in Figure 1.

ITS station management is specified as a distributed process, where no supervisory entity is employed.

Intelligent transport systems — ITS station management —

Part 4:
Station-internal management communications

1 Scope

This document provides specifications for secure ITS station-internal management communications.

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/IEC 8825-2, *Information technology — ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)*

ISO 17419:2018, *Intelligent transport systems — Cooperative ITS — Globally unique identification*

ISO 21217:2014, *Intelligent transport systems — Communications access for land mobiles (CALM) — Architecture*

ISO 21218, *Intelligent transport systems — Hybrid communications — Access technology support*

ISO 24102-1, *Intelligent transport systems — ITS station management — Part 1: Local management*

ISO 24102-3, *Intelligent transport systems — ITS station management — Part 3: Service access points*

ISO 24102-6, *Intelligent transport systems — ITS station management — Part 6: Path and flow management*

ETSI TS 102 797-2, *Intelligent Transport Systems (ITS); Road Transport and Traffic Telematics (RTTT); Test specifications for Intelligent Transport Systems, Communications access for land mobiles (CALM), ITS station management (ISO 24102); Part 2: Test Suite Structure and Test Purposes (TSS & TP)*

ETSI TS 102 797-3, *Intelligent Transport Systems (ITS); Road Transport and Traffic Telematics (RTTT); Test specifications for Intelligent Transport Systems, Communications access for land mobiles (CALM), ITS station management (ISO 24102); Part 3: Abstract Test Suite (ATS) and partial PIXIT information*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 21217, ISO 21218, ISO 24102-1, and ISO 24102-3 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1
ITS-S communication unit
physical unit in an ITS-SU containing a part or all of the functionality of an ITS-S

[SOURCE: ISO 21217:2014, 3.21, modified — Note 1 to entry was deleted.]

4 Symbols and abbreviated terms

For the purposes of this document, the abbreviated terms given in ISO 21217:2014, ISO 21218, ISO 24102-1, and ISO 24102-3 and the following apply.

ITS-SCU	ITS station Communication Unit
ITS-SCU-ID	ITS-SCU Identifier
ITS-SCUID	Globally unique identifier of an ITS-SCU

NOTE ITS-SCUID is specified in ISO 17419 as a globally unique identifier, whilst ITS-SCU-ID is specified in this document as an addressing element for IIC.

IIC	ITS station-internal management communications
IICM	IIC manager
IICA	IIC agent
IICP	IIC protocol
n/a	not applicable

5 ITS station management

The ITS station management includes functionality specified in the various parts of this multi-part International Standard:

1) The functionality of local ITS station management specified in ISO 24102-1.
2) The functionality of remote ITS station management specified in ISO 24102-2[2].
3) The functionality of service access points specified in ISO 24102-3.
4) **The functionality of ITS station-internal management communications specified in this document (Part 4).**
5) Void.
6) The functionality of path and flow management specified in ISO 24102-6.

ITS station-internal management communications interconnects ITS station communication units (ITS-SCUs) of the same ITS station (ITS-S) via the ITS station-internal network illustrated in ISO 21217. This communication is also referred to as "ITS-S Internal management Communications" (IIC) in this document. IIC allows remote access to management SAPs specified in ISO 21217 with details specified in ISO 24102-3.

IIC may be secured following the principles of trusted distributed systems.

Detailed mandatory requirements are specified in the following clauses of this document.

— Clause 6 specifies the IIC reference architecture.
— Clause 7 specifies IIC protocol data units (PDUs).
— Clause 8 specifies communication procedures.
— Clause 9 specifies management procedures.
— Clause 10 specifies security elements and procedures.

— Clause 11 specifies conformance declaration.

— Clause 12 specifies test methods.

— Annex A specifies an ASN.1 module.

— Annex B specifies IIC PDUs.

— Annex C specifies the implementation conformance statement (ICS) proforma.

— Annex D exemplifies communication service parameters for IIC for usage in implementations compliant with ISO 17423[5].

6 Reference architecture

"ITS-S Internal management Communications" (IIC) is communications between ITS-S Management Entities of different ITS-SCUs of the same ITS-SU via the ITS station-internal network. A specific purpose of IIC is remote access to management service access points MI-SAP, MN-SAP, MF-SAP, MA-SAP and MS-SAP. The concept of ITS-SCUs is specified in ISO 21217. An ITS-SCU can support IICP disregard of the ITS-S roles it supports.

NOTE The MA-SAP is part of the API presented in Figure 2.

The reference architecture for IIC is illustrated in Figure 2.

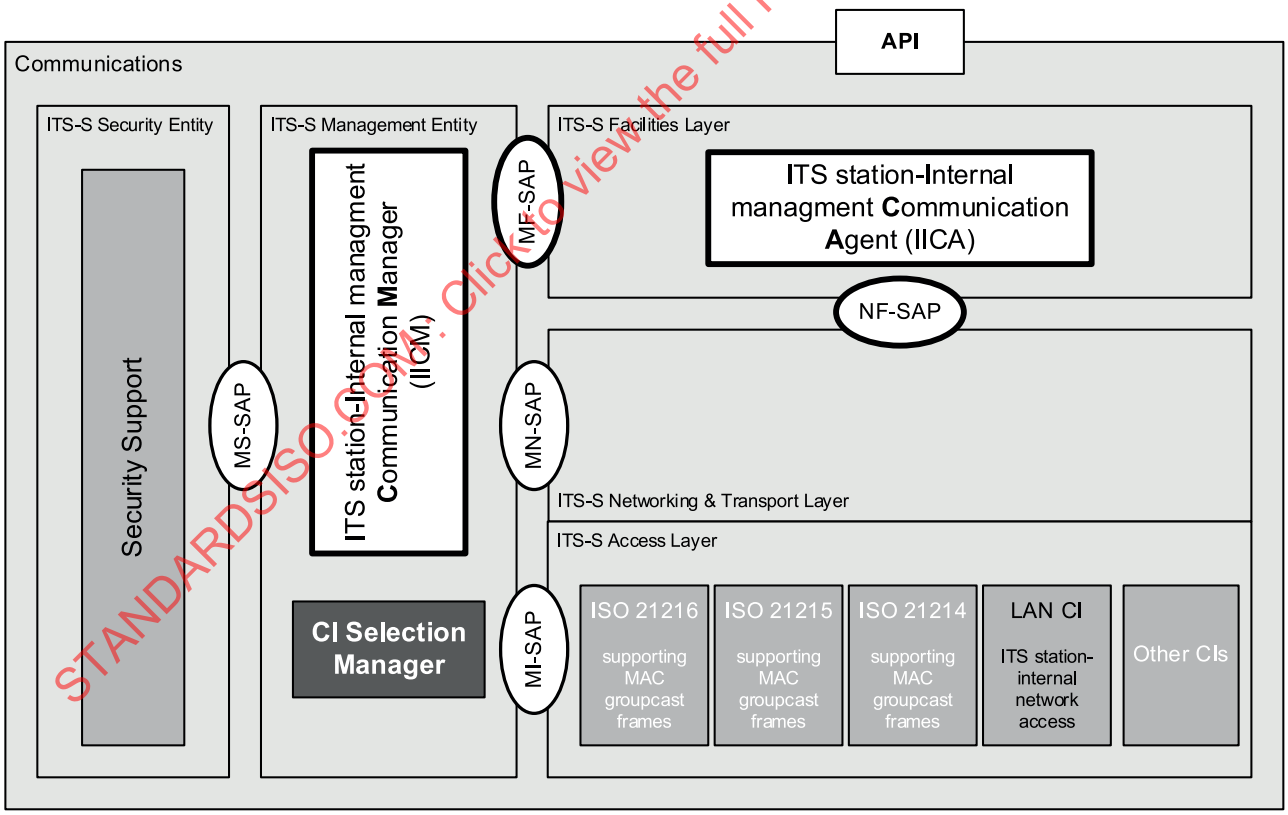

Figure 2 — Reference architecture for ITS station-internal management communications

Source and destination of IIC protocol data units are instances of the "ITS-S Internal management Communications Manager" (IICM). Transmission requests from the IICM are forwarded to the "ITS-S Internal management Communications Agent" (IICA) via the MF-SAP. Notifications of received IIC PDUs are sent by the IICA via the MF-SAP to the IICM.

Communications on the ITS station-internal network is performed between peer instances of the IICA via NF-SAP, a suitable networking & transport layer protocol, the IN-SAP, and a suitable access technology. Source- and destination endpoints of the ITS-S networking & transport layer protocol are identified by an ITS-S port (ITS-SP) with the number PORT_IIC of the IICA identified in [3].

7 Protocol data units

"ITS-S Internal management Communications" (IIC) uses the following protocol data units (PDU) illustrated in Figure 3:

— IIC-Request;

— IIC-Response.

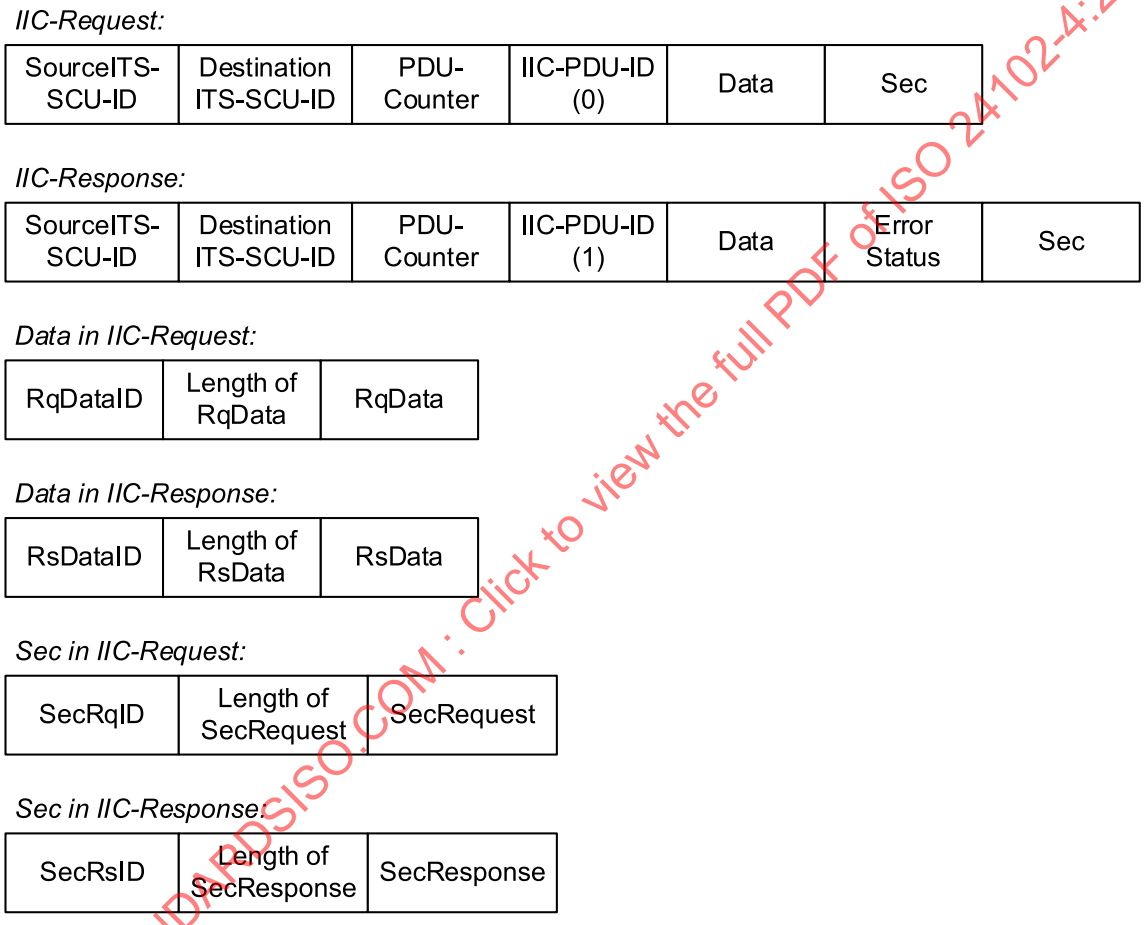

Figure 3 — IIC PDU structure

The IIC-Request PDU is of ASN.1 type `IIC-Request` specified in A.2.

The IIC-Response PDU is of ASN.1 type `IIC-Response` specified in A.2.

Details on parameters of these PDUs are specified in Table 1.

Table 1 — IIC PDUs

PDU element/ASN.1 type	IIC-Request	IIC-Response
Source ITS-SCU-ID/ `ITS-scuId`	ITS-SCU-ID of source ITS-SCU, which produces the request.	ITS-SCU-ID of ITS-SCU, which produces the response.
Destination ITS-SCU-ID/ `ITS-scuId`	ITS-SCU-ID of destination ITS-SCU.	Same as SourceITS-SCU-ID of related request if not requested otherwise in this document.
PDU-Counter/ `PduCounter`	Even number generated from a cyclic counter at the ITS-SCU, which produces the request.	PDU-Counter of related request incremented by one.
IIC-PDU-ID / `INTEGER` **Value dependent on PDU type**	One octet INTEGER value 0. ASN.1 type `RequestID`.	One octet INTEGER value 1. ASN.1 type `ResponseID`.
Data/ `SEQUENCE` **based on ASN.1** `CLASS IICPREQ`	Data type identifier RqDataID followed by length of RqData in Integer multiples of an octet and RqData. ASN.1 type `PduRequest`.	Data type identifier RsDataID followed by length of RsData in Integer multiples of an octet and RsData ASN.1 type `PduResponse`.
ErrorStatus/ `PduErrStatus`	Not existent.	Existent. 0: No error happened >0: Number indicating type of error.
Seq/ `SEQUENCE` **based on ASN.1** `CLASS SECIICP`	Security type identifier SecRqID followed by the lenth of SecRequest in Integer multiples of an octet and SecRequest. ASN.1 type `SecRq`.	Security type identifier SecRsID followed by the lenth of SecResponse in Integer multiples of an octet and SecResponse. ASN.1 type `SecRs`.

The ASN.1 specification of the PDUs as provided in Annex A of this document shall apply.

ITS-SCU-ID values used in parameters "SourceITS-SCU-ID" and "DestinationITS-SCU-ID" shall be as specified in Table 2.

Table 2 — ITS-SCU-ID value assignment

SourceITS-SCU-ID	DestinationITS-SCU-ID	Description
0		Reserved. Used to indicate "own/local ITS-SCU". Must not be used in communications with other ITS-SCUs.
n/a	1	ITS-SCU-ID identifying ITS-SCUs with ITS-S host role.
n/a	2	ITS-SCU-ID identifying ITS-SCUs with ITS-S router role.
3		ITS test system Dispatcher (using IICP for upper tester access), see ISO/TS 20026[4].
4		ITS Test CI unit, see ISO/TS 20026[4].
5 to 15		Reserved. ITS-SCU-ID identifying ITS-SCUs with an implementation specific role
16 to 65534		ITS-SCU-ID identifying uniquely a specific ITS-SCU in an ITS station.
n/a	65535	ITS-SCU-ID identifying all ITS-SCUs.

"Data" values shall be set as specified in Annex B of this document.

"ErrorStatus" values shall be set as specified in Table 3.

Table 3 — ErrorStatus value assignment; ASN.1 type `PduErrStatus`

ErrorStatus	Description
0	No error
1	IIC-PDU-ID unknown or not implemented
2	Duplicate ITS-SCU-ID
3	Invalid or unknown AliveMessage
4	Invalid or unknown ITS-SCU type
5 to 127	Reserved for future use
128 to 254	For implementation-specific, non-standardized usage
255	Unspecified error

The ITS port number PORT_IICP is specified in ISO 17419: to identify the IICA. Port numbers for TCP-IP or UDP-IP are not allocated so far.

8 Communication procedures

8.1 Initialization

8.1.1 IICM

ITS station-internal management communications between addressable ITS-SCUs shall be initialized as specified in Clause 9 on management procedures.

8.1.2 IICA

Prior to the initialization specified in Clause 9, the IICA shall initialize communications as requested for the selected ITS-S networking & transport layer protocol.

8.2 Transmission

8.2.1 IIC-Request PDU

Upon request from a protocol in the ITS-S management entity, the IICM shall construct the IIC-Request PDU specified in Clause 7. In case secure transmission is needed, the "Security Support" illustrated in Figure 2 shall be involved via the MS-SAP. Details on security shall be as specified in Clause 10.

The "PDU-Counter" value shall be set to an even value uniquely in an ITS-SCU under the control of the IICM issuing the request. The initial value shall be zero. For every next IIC-Request PDU, the counter value shall be incremented by two. It shall wrap from 65534 to zero. The IICM shall note the value of the PDU-Counter in case a response is expected.

The IICM shall forward the IIC-Request PDU to the IICA using MF-COMMAND `IICrequestTX` specified in Annex A.

The IICA shall request transmission of the IIC-Request PDU using the appropriate service of the NF-SAP.

8.2.2 IIC-Response PDU

Upon reception of an IIC-Request PDU which requires transmission of an IIC-Response PDU, the IICM shall construct the IIC-Response PDU specified in Clause 7. In case secure transmission is needed, the "Security support" illustrated in Figure 2 shall be involved via the MS-SAP. Details on security shall be as specified in Clause 10.

The "PDU-Counter" value shall be set equal to the value of the "PDU-Counter" contained in the related IIC-Request PDU incremented by one.

The IICM shall forward the IIC-Response PDU to the IICA using MF-COMMAND `IICresponseTX` specified in Annex A.

The IICA shall request transmission of the IIC-Response PDU using the appropriate service of the NF-SAP.

8.3 Reception

8.3.1 Initial processing at the IICA

The IICA shall perform the following steps:

1) Check DestinationITS-SCU-ID:

   i) If the DestinationITS-SCU-ID value is a valid address value (3, 4, 16 to 65534) and matches the own address, step 2) shall be performed.

   ii) If the DestinationITS-SCU-ID value is either 1 or 2, and the value matches with the own ITS-S role, step 2) shall be performed.

   iii) If the DestinationITS-SCU-ID value is 65535, step 2) shall be performed.

   iv) Otherwise the PDU is ignored.

2) Check the PDU type value and PDU-Counter value.

   i) In case the PDU type is inconsistent with the PDU-Counter value, the PDU is ignored.

   ii) The IICA shall forward an IIC-Request PDU received via NF-SAP to the IICM using MF-REQUEST `IICrequestRX` specified in Annex A.

   iii) The IICA shall forward an IIC-Response PDU received via NF-SAP to the IICM using MF-REQUEST `IICresponseRX` specified in Annex A.

8.3.2 Initial processing at the IICM

The IICM shall perform the following steps:

1) In case the element SecRq is not empty, involve the "Security Support" illustrated in Figure 2 via the MS-SAP as specified in Clause 10.

2) In case of unsuccessful security checking, the IICM ignores the received PDU.

3) In case of successful security checking, the IICM shall perform the respective procedure for IIC-Request or IIC-Response.

8.3.3 Final IIC-Request PDU procedure at the IICM

The IICM shall perform the following steps:

1) The IICM performs the required action as indicated by Data contained in the IIC-Request PDU.

2) In case a response is required, the IICM prepares the IIC-Response PDU as specified above.

8.3.4 IIC-Response PDU procedure

The IICM shall perform the following steps:

1) Check whether the PDU-Counter value matches a previously transmitted IIC-Request that was not acknowledged so far with an IIC-Response PDU. In case of no match the IICM ignores the received PDU.

2) The IICM performs the required action as indicated by Data contained in the IIC-Response PDU.

9 Management procedures

9.1 General

The management procedures specified in Clause 9 include procedures

— for initial assignment of unique ITS-SCU-IDs,

— for ITS-SCU-ID maintenance, and

— for release of ITS-SCU-IDs.

Uniqueness of ITS-SCU-IDs in the range 15 to 65534 may be achieved by implementation-dependent means.

9.2 ITS-SCU-ID assignment

In case an ITS-SCU does not have a pre-defined unique ITS-SCU-ID, the procedure specified in this sub-clause shall be followed.

The IICM shall generate an ITS-SCU-ID as specified in Table 2 as its own ITS-SCU-ID and shall put the selected ITS-SCU-ID to its local ITS-SCU-list. ITS-SCU-ID values already being in the local ITS-SCU-list indicate usage by another ITS-SCU in the same station and shall not be selected. Then the IIC-Request PDU `ITS-SCUalive` (new) shall be sent to all ITS-SCUs, indicating a first choice of ITS-SCU-ID in the "SourceITS-SCU-ID element, or a new choice of ITS-SCU-ID, and the type of ITS-SCU in the "Data" element. In case of a negative acknowledgement, i.e. indication of usage of this ITS-SCU-ID value by another ITS-SCU, see below, the IICM shall repeat the procedure with a new ITS-SCU-ID value.

Upon activation, the IICM may listen to receive IIC-Request PDUs, or IIC-Response PDUs in order to identify already allocated ITS-SCU-IDs.

Upon reception of an IIC-Request PDU `ITS-SCUalive` (new), the IICM shall check the SourceITS-SCU-ID.

— If the SourceITS-SCU-ID is equal to the own ITS-SCU-ID, the receiving IICM shall send an IIC-Response PDU `ITS-SCUalive` to all ITS-SCUs, reporting the own ITS-SCU-ID and type of ITS-SCU to all ITS-SCUs, indicating ErrorStatus=2.

— If the SourceITS-SCU-ID is different to the own ITS-SCU-ID, the IICM shall take this information to its local ITS-SCU-list, if not already present there. An existing entry shall not be updated with this new information. The IICM shall acknowledge the IIC-Request PDU with the IIC-Response PDU `ITS-SCUalive`, reporting the own ITS-SCU-ID and type of ITS-SCU, indicating ErrorStatus=0.

Upon reception of an IIC-Request PDU `ITS-SCUalive` (alive), the IICM shall check the SourceITS-SCU-ID.

— If the SourceITS-SCU-ID is equal to the own ITS-SCU-ID, the receiving IICM shall send an IIC-Response PDU `ITS-SCUalive` to all ITS-SCUs, reporting the own ITS-SCU-ID and type of ITS-SCU to all ITS-SCUs, indicating ErrorStatus=2. All IICMs with this ITS-SCU-ID shall invalidate this ITS-SCU-ID and shall start the procedure to select a new ITS-SCU-ID.

— If the SourceITS-SCU-ID is different to the own ITS-SCU-ID, the IICM shall take this information to its local ITS-SCU-list, if not already present there. The IIC-Request PDU shall not be acknowledged with an IIC-Response.

Upon reception of an IIC-Response PDU `ITS-SCUalive`, the IICM shall check the ErrorStatus.

— In case of ErrorStatus=0, the IICM shall take this information about another ITS-SCU to its local ITS-SCU-list, if not already present there.

— In case of ErrorStatus=2, an address conflict was detected. The procedure to be selected upon this event depends on the value of "Message" and SourceITS-SCU-ID contained in the IIC-Response as presented in Table 4.

Table 4 — Error handling procedure for IIC-Response PDU (`ITS-SCUalive`)

ErrorStatus = 2 (Duplicate ITS-SCU-ID)	SourceITS-SCU-ID = own ITS-SCU-ID	SourceITS-SCU-ID ≠ own ITS-SCU-ID
AliveMessage = new	Map ITS-SCU-ID to ITS-SCUtype reported in the response. Restart ITS-SCU-ID assignment process.	Nothing to do.
AliveMessage = alive	**IMPORTANT — This should never happen.** Reset ITS-SCU and restart ITS-SCU-ID assignment process.	**IMPORTANT — This should never happen.** Delete SourceITS-SCU-ID from local ITS-SCU-list.
AliveMessage = delete	**IMPORTANT — This should never happen.** Nothing to do.	

9.3 Maintenance of ITS-SCU-ID

The IICM shall periodically transmit the "alive-signal" IIC-Request PDU `ITS-SCUalive` (alive) in order to indicate its presence in the ITS station. The period of transmission shall be as set in parameter "Talive" specified in ISO 24102-1. The value of "Talive" shall be defined by implementation and shall be unique in an ITS station.

The IICM periodically shall check the local ITS-SCU-list. If for a period of at least three times "Talive" no "alive-signal" IIC-Request PDU `ITS-SCUalive` (alive) was received, the IICM assumes that this ITS-SCU is no longer alive, and the respective ITS-SCU-ID shall be deleted from the local ITS-SCU-list.

9.4 Shut-down of ITS-SCU

In case an ITS-SCU has the capability to perform a power shut-down, prior to performing such a shut-down of an ITS-SCU, the IIC-Request PDU `ITS-SCUalive` (delete) shall be sent to all ITS-SCUs at least once. This message shall not be acknowledged.

10 Security

Details of security data elements presented in Figure 3, and related security procedures are out of scope of this document.

11 Conformance

The "Protocol Implementation Conformance Statements" (PICS) proforma specified in Annex C supersedes the one specified in ETSI TS 102 797-1[7] for the predecessor of this document.

12 Test methods

The "Test Suite Structure & Test Purposes" (TSS&TP) for conformance testing can be based on ETSI TS 102 797-2, which is to be cross-checked for identifying potential but not expected inconsistencies introduced by the revision of the predecessor of this document.

The "Abstract Test Suite" (ATS) for conformance testing can be based on ETSI TS 102 797-3, which needs to be revised due to changes of ASN.1 type names and ASN.1 value names.

Annex A
(normative)

ASN.1 module

A.1 Overview

The following ASN.1 module is specified in this Annex:

— ITSiitsscu { iso (1) standard (0) calm-management (24102) iitsscu (4) asnm-1 (1) version2 (2)}

In case the ASN.1 specifications given in this Annex are not compliant with illustrations or specifications provided elsewhere in this document, the specifications given in this Annex shall prevail.

A.2 Module ITSiitsscu

This module specifies ASN.1 type definitions together with useful ASN.1 value definitions.

Unaligned packed encoding rules (PER) as specified in ISO/IEC 8825-2 shall be applied for this ASN.1 module.

In order to achieve octet alignment enabling cheap implementations, "fill" bits were defined. All fill bits shall be set to the value '0'b.

```
ITSiitsscu { iso (1) standard (0) calm-management (24102) iitsscu (4)
asnm-1 (1) version2 (2)}
DEFINITIONS AUTOMATIC TAGS::=BEGIN

IMPORTS
-- From EN ISO 17419-1
ITS-scuId, ITSatt FROM CITSapplMgmtApplReg2 {iso(1) standard(0)
cits-applMgmt (17419) applRegistry (2) version2 (2)}

-- From EN ISO 17419-1 Data Dictionary
NullType FROM CITSdataDictionary1 {iso(1) standard(0) cits-applMgmt
(17419) dataDictionary (1) version1 (1)}

-- From ISO 21218:2017
CIaClass, CIclass, CIstatus, Directivity, Link-ID, MACaddress FROM
ITSllsap {iso(1) standard(0) calm-ll-sap(21218) asnm-1 (1) version2
(2)}

RefMPARAM, Param24102 FROM ITSmanagement { iso (1) standard (0)
calm-management (24102) local (1) asnm-1 (1)  version2 (2)}

-- From ISO 24102-3 ITSsapCmdMf
SimFUTcmd, SimFLTcmd, MF-Command-request, MF-Command-confirm FROM
ITSsapCmdMf { iso (1) standard (0) calm-management (24102) msap (3)
commands  (3) mf (2) version0 (0)}

-- From ISO 24102-3 ITSsapCmdMn
SimNUTcmd, SimNLTcmd, MN-Command-request, MN-Command-confirm FROM
ITSsapCmdMn { iso (1) standard (0) calm-management (24102) msap (3)
commands  (3) mn (3) version0 (0)}

-- From ISO 24102-3 ITSsapCmdMi
SimIUTcmd, MI-Command-request, MI-Command-confirm, MI-Get-request,
MI-Set-request, MI-Get-confirm, MI-Set-confirm FROM ITSsapCmdMi
{ iso (1) standard (0) calm-management (24102) msap (3) commands
(3) mi (4) version0 (0)}
```

```
-- From ISO 24102-3 ITSsapReqMf
SimFLTreq, SimFUTreq, MF-Request-request, MF-Request-confirm FROM
ITSsapReqMf { iso (1) standard (0) calm-management (24102) msap
(3) requests  (4)  mf (2) version0 (0)}

-- From ISO 24102-3 ITSsapReqMn
SimNUTreq, SimNLTreq, MN-Request-request, MN-Request-confirm FROM
ITSsapReqMn { iso (1) standard (0) calm-management (24102) msap (3)
requests  (4)  mn (3) version0 (0)}

-- From ISO 24102-3 ITSsapReqMi
SimIUTreq, MI-Request-request, MI-Request-confirm FROM ITSsapReqMi
{ iso (1) standard (0) calm-management (24102) msap (3) requests
(4) mi (4) version0 (0)}

-- From ISO 24102-3 ITSsapBasics
ErrStatus FROM ITSsapBasics { iso (1) standard (0) calm-management
(24102) msap (3) basics (2) version0 (0)}

-- From ISO 24102-3 ITSsapCmdMa
MA-Command-request, MA-Command-confirm FROM ITSsapCmdMa { iso (1)
standard (0) calm-management (24102) msap (3) commands  (3) ma (1)
version0 (0)}

-- From ISO 24102-3 ITSsapReqMa
MA-Request-request, MA-Request-confirm FROM ITSsapReqMa { iso (1)
standard (0) calm-management (24102) msap (3) requests  (4)  ma (1)
version0 (0)}

-- From ISO 24102-3 ITSsapCmdMs
MS-Command-request, MS-Command-confirm FROM ITSsapCmdMs { iso (1)
standard (0) calm-management (24102) msap (3) commands  (3) ms (0)
version0 (0)}

-- From ISO 24102-3 ITSsapReqMs
MS-Request-request, MS-Request-confirm FROM ITSsapReqMs { iso (1)
standard (0) calm-management (24102) msap (3) requests  (4)  ms (0)
version0 (0)}

-- Details on SecRq and SecRs to be provided by a standard on security
;
-- End of IMPORTS

-- Types

-- PDUs --
-- Data type identifier for IIC-Request and IIC-Response
RefIICP ::= INTEGER { -- Define IICPREQ.ref values range
   c-iic-alive          (0),
   c-iic-mfRcmd         (1),
   c-iic-mfRreq         (2),
   c-iic-mnRcmd         (3),
   c-iic-mnRreq         (4),
   c-iic-miRcmd         (5),
   c-iic-miRreq         (6),
   c-iic-miRget         (7),
   c-iic-miRset         (8),
   c-iic-mCmd           (9),
   c-iic-maRcmd         (10),
   c-iic-maRreq         (11),
   c-iic-msRcmd         (12),
   c-iic-msRreq         (13),
-- c-iic-simSutCmd      (236),
-- c-iic-simSutReq      (237),
-- c-iic-simSltCmd      (238),
-- c-iic-simSltReq      (239),
   c-iic-simUtIn        (240), -- see ETSI abstract test suite
   c-iic-simUtOut       (241), -- see ETSI abstract test suite
-- c-iic-simAutCmd      (242),
-- c-iic-simAutReq      (243),
   c-iic-simFutCmd      (244),
```

```
   c-iic-simFutReq      (245),
   c-iic-simNutCmd      (246),
   c-iic-simNutReq      (247),
   c-iic-simIutCmd      (248),
   c-iic-simIutReq      (249),
-- c-iic-simAltCmd      (250),
-- c-iic-simAltReq      (251),
   c-iic-simFltCmd      (252),
   c-iic-simFltReq      (253),
   c-iic-simNltCmd      (254),
   c-iic-simNltReq      (255)
} (0..255)

-- IIC-Request PDU
IIC-Request::=SEQUENCE{
   sourceITS-scuId         ITS-scuId,
   destinationITS-scuId    ITS-scuId,
   pduCounter              PduCounter,
   requestID               RequestID, -- value zero
   pduRequest              PduRequest,
   secRq                   SecRq
   }

RequestID::=INTEGER{
   request      (0)
   } (0..255)

IICPREQ::=CLASS{
   &ref RefIICP UNIQUE, -- data type identifier
   &IICPrequest
   }

IICPrequests IICPREQ::={aliveRq | mfRcmdRq | mfRreqRq | mnRcmdRq |
mnRreqRq | miRcmdRq | miRreqRq | miRgetRq | miRsetRq | mCmdRq |
maRcmdRq | maRreqRq | msRcmdRq | msRreqRq | simFutCmdRq | simFutReqRq |
simNutCmdRq | simNutReqRq | simIutCmdRq | simIutReqRq | simFltCmdRq |
simFltReqRq | simNltCmdRq | simNltReqRq | simUtInRq | simUtOutRq, ...}

PduRequest::=SEQUENCE{
   reqRef    IICPREQ.&ref({IICPrequests}), -- data type identifier
   iicpReq   IICPREQ.&IICPrequest({IICPrequests}{@reqRef})
   }

-- Used in operational mode
aliveRq     IICPREQ::={&ref c-iic-alive, &IICPrequest ITS-SCUalive}
mfRcmdRq    IICPREQ::={&ref c-iic-mfRcmd, &IICPrequest MF-Command-
            request}
mfRreqRq    IICPREQ::={&ref c-iic-mfRreq, &IICPrequest MF-Request-
            request}
mnRcmdRq    IICPREQ::={&ref c-iic-mnRcmd, &IICPrequest MN-Command-
            request}
mnRreqRq    IICPREQ::={&ref c-iic-mnRreq, &IICPrequest MN-Request-
            request}
miRcmdRq    IICPREQ::={&ref c-iic-miRcmd, &IICPrequest MI-Command-
            request}
miRreqRq    IICPREQ::={&ref c-iic-miRreq, &IICPrequest MI-Request-
            request}
miRgetRq    IICPREQ::={&ref c-iic-miRget, &IICPrequest MI-Get-request}
miRsetRq    IICPREQ::={&ref c-iic-miRset, &IICPrequest MI-Set-request}
mCmdRq      IICPREQ::={&ref c-iic-mCmd, &IICPrequest McmdRq}
maRcmdRq    IICPREQ::={&ref c-iic-maRcmd, &IICPrequest MA-Command-
            request}
maRreqRq    IICPREQ::={&ref c-iic-maRreq, &IICPrequest MA-Request-
            request}
msRcmdRq    IICPREQ::={&ref c-iic-msRcmd, &IICPrequest MS-Command-
            request}
msRreqRq    IICPREQ::={&ref c-iic-msRreq, &IICPrequest MS-Request-
            request}
-- Used in test mode only
-- simAutCmdRq    IICPREQ::={&ref c-iic-simAutCmd, &IICPrequest
                  SimAUTcmd}
```

```
-- simAutReqRq    IICPREQ::={&ref c-iic-simAutReq, &IICPrequest
                  SimAUTreq}
simFutCmdRq       IICPREQ::={&ref c-iic-simFutCmd, &IICPrequest
                  SimFUTcmd}
simFutReqRq       IICPREQ::={&ref c-iic-simFutReq, &IICPrequest
                  SimFUTreq}
simNutCmdRq       IICPREQ::={&ref c-iic-simNutCmd, &IICPrequest
                  SimNUTcmd}
simNutReqRq       IICPREQ::={&ref c-iic-simNutReq, &IICPrequest
                  SimNUTreq}
simIutCmdRq       IICPREQ::={&ref c-iic-simIutCmd, &IICPrequest
                  SimIUTcmd}
simIutReqRq       IICPREQ::={&ref c-iic-simIutReq, &IICPrequest
                  SimIUTreq}
-- simAltCmdRq    IICPREQ::={&ref c-iic-simAltCmd, &IICPrequest
                  SimALTcmd}
-- simAltReqRq    IICPREQ::={&ref c-iic-simAltReq, &IICPrequest
                  SimALTreq}
simFltCmdRq       IICPREQ::={&ref c-iic-simFltCmd, &IICPrequest
                  SimFLTcmd}
simFltReqRq       IICPREQ::={&ref c-iic-simFltReq, &IICPrequest
                  SimFLTreq}
simNltCmdRq       IICPREQ::={&ref c-iic-simNltCmd, &IICPrequest
                  SimNLTcmd}
simNltReqRq       IICPREQ::={&ref c-iic-simNltReq, &IICPrequest
                  SimNLTreq}
simUtInRq         IICPREQ::={&ref c-iic-simUtIn, &IICPrequest SimUt}
simUtOutRq        IICPREQ::={&ref c-iic-simUtOut, &IICPrequest SimUt}

ITS-SCUalive::=SEQUENCE{
   message        AliveMessage,
   its-scuType    ITS-SCUtype
   }

AliveMessage::= INTEGER{
   alive     (0),
   delete    (1),
   new       (255)
   } (0..255)

ITS-SCUtype::= INTEGER{
   unknown          (0),
   host             (1),
   router           (2),
   itsTestSystem    (3),
   testCIunit       (4),
   hostAndRouter    (5),
   any              (255)
   } (0..255)


-- SimUt CLASS
RefSimUt ::= INTEGER { -- allows for 256 commands
   c-simUt-mfCmdReq     (0),
   c-simUt-mfCmdConf    (1),
   c-simUt-mnCmdReq     (2),
   c-simUt-mnCmdConf    (3),
   c-simUt-miCmdReq     (4),
   c-simUt-miCmdConf    (5),
   c-simUt-maCmdReq     (6),
   c-simUt-maCmdConf    (7),
   c-simUt-mfReqReq     (8),
   c-simUt-mfReqConf    (9),
   c-simUt-mnReqReq     (10),
   c-simUt-mnReqConf    (11),
   c-simUt-miReqReq     (12),
   c-simUt-miReqConf    (13),
   c-simUt-maReqReq     (14),
   c-simUt-maReqConf    (15),
   c-simUt-miGetReq     (16),
   c-simUt-miGetConf    (17),
```

```
   c-simUt-miSetReq     (18),
   c-simUt-miSetConf    (19),
   c-simUt-mcmdReq      (20),
   c-simUt-mcmdConf     (21),
   c-simUt-msCmdReq     (22),
   c-simUt-msCmdConf    (23),
   c-simUt-msReqReq     (24),
   c-simUt-msReqConf    (25)
   } (0..255)

SIMUTC::=CLASS{
   &refSimUt RefSimUt UNIQUE, -- allows for 256 commands
   &SimUtCmd
   }

SimUt::=SEQUENCE{
   simUtRef    SIMUTC.&refSimUt({SimUts}),
   simUt       SIMUTC.&SimUtCmd ({SimUts}{@simUtRef})
   }


SimUts SIMUTC::={simUt-maCmdConf | simUt-maCmdReq | simUt-maReqConf |
simUt-maReqReq | simUt-mcmdConf | simUt-mcmdReq | simUt-mfCmdConf |
simUt-mfCmdReq | simUt-mfReqConf | simUt-mfReqReq | simUt-miCmdConf |
simUt-miCmdReq | simUt-miGetConf | simUt-miGetReq | simUt-miReqConf |
simUt-miReqReq | simUt-miSetConf | simUt-miSetReq | simUt-mnCmdConf |
simUt-mnCmdReq | simUt-mnReqConf | simUt-mnReqReq | simUt-msCmdConf |
simUt-msCmdReq | simUt-msReqConf | simUt-msReqReq, ...}

simUt-mfCmdReq     SIMUTC::={&refSimUt c-simUt-mfCmdReq, &SimUtCmd
                   MF-Command-request}
simUt-mfCmdConf    SIMUTC::={&refSimUt c-simUt-mfCmdConf, &SimUtCmd
                   MF-Command-confirm}
simUt-mnCmdReq     SIMUTC::={&refSimUt c-simUt-mnCmdReq, &SimUtCmd
                   MN-Command-request}
simUt-mnCmdConf    SIMUTC::={&refSimUt c-simUt-mnCmdConf, &SimUtCmd
                   MN-Command-confirm}
simUt-miCmdReq     SIMUTC::={&refSimUt c-simUt-miCmdReq, &SimUtCmd
                   MI-Command-request}
simUt-miCmdConf    SIMUTC::={&refSimUt c-simUt-miCmdConf, &SimUtCmd
                   MI-Command-confirm}
simUt-maCmdReq     SIMUTC::={&refSimUt c-simUt-maCmdReq, &SimUtCmd
                   MA-Command-request}
simUt-maCmdConf    SIMUTC::={&refSimUt c-simUt-maCmdConf, &SimUtCmd
                   MA-Command-confirm}
simUt-mfReqReq     SIMUTC::={&refSimUt c-simUt-mfReqReq, &SimUtCmd
                   MF-Request-request}
simUt-mfReqConf    SIMUTC::={&refSimUt c-simUt-mfReqConf, &SimUtCmd
                   MF-Request-confirm}
simUt-mnReqReq     SIMUTC::={&refSimUt c-simUt-mnReqReq, &SimUtCmd
                   MN-Request-request}
simUt-mnReqConf    SIMUTC::={&refSimUt c-simUt-mnReqConf, &SimUtCmd
                   MN-Request-confirm}
simUt-miReqReq     SIMUTC::={&refSimUt c-simUt-miReqReq, &SimUtCmd
                   MI-Request-request}
simUt-miReqConf    SIMUTC::={&refSimUt c-simUt-miReqConf, &SimUtCmd
                   MI-Request-confirm}
simUt-maReqReq     SIMUTC::={&refSimUt c-simUt-maReqReq, &SimUtCmd
                   MA-Request-request}
simUt-maReqConf    SIMUTC::={&refSimUt c-simUt-maReqConf, &SimUtCmd
                   MA-Request-confirm}
simUt-miGetReq     SIMUTC::={&refSimUt c-simUt-miGetReq, &SimUtCmd
                   MI-Get-request}
simUt-miGetConf    SIMUTC::={&refSimUt c-simUt-miGetConf, &SimUtCmd
                   MI-Get-confirm}
simUt-miSetReq     SIMUTC::={&refSimUt c-simUt-miSetReq, &SimUtCmd
                   MI-Set-request}
simUt-miSetConf    SIMUTC::={&refSimUt c-simUt-miSetConf, &SimUtCmd
                   MI-Set-confirm}
simUt-mcmdReq      SIMUTC::={&refSimUt c-simUt-mcmdReq, &SimUtCmd
                   McmdRq}
```

```
simUt-mcmdConf     SIMUTC::={&refSimUt c-simUt-mcmdConf, &SimUtCmd
                   McmdRs}
simUt-msCmdReq     SIMUTC::={&refSimUt c-simUt-msCmdReq, &SimUtCmd
                   MS-Command-request}
simUt-msCmdConf    SIMUTC::={&refSimUt c-simUt-msCmdConf, &SimUtCmd
                   MS-Command-confirm}
simUt-msReqReq     SIMUTC::={&refSimUt c-simUt-msReqReq, &SimUtCmd
                   MS-Request-request}
simUt-msReqConf    SIMUTC::={&refSimUt c-simUt-msReqConf, &SimUtCmd
                   MS-Request-confirm}

-- General OBJECT CLASS for management commands
RefMcmd ::= INTEGER { -- allows for 256 commands
   c-mcmd-ping             (0),
   c-mcmd-vCIinfo          (1),
   c-mcmd-vCIupdate        (2),
   c-mcmd-getParam24102    (3),
   c-mcmd-setParam24102    (4),
   c-mcmd-testIicp         (254),
   c-mcmd-testConfig      (255)
} (0..255)

MCMD::=CLASS{
   &refMcmd RefMcmd UNIQUE, -- allows for 256 commands
   &Mcmd
   }

MCMDREQ::=MCMD

McmdRq::=SEQUENCE{
   mCmdRef    MCMDREQ.&refMcmd({McmdReqs}),
   mcmd       MCMDREQ.&Mcmd ({McmdReqs}{@mCmdRef})
   }

McmdReqs MCMDREQ::={pingRq | vCIinfoRq | vCIupdateRq | getParam24102Rq |
setParam24102Rq | testIicpRq | testConfigRq, ...}

pingRq             MCMDREQ::={&refMcmd c-mcmd-ping, &Mcmd PingIICP}
vCIinfoRq          MCMDREQ::={&refMcmd c-mcmd-vCIinfo, &Mcmd
                   VCI-info-req}
vCIupdateRq        MCMDREQ::={&refMcmd c-mcmd-vCIupdate, &Mcmd
                   VCI-update-req}
getParam24102Rq    MCMDREQ::={&refMcmd c-mcmd-getParam24102, &Mcmd
                   GetParam24102Rq}
setParam24102Rq    MCMDREQ::={&refMcmd c-mcmd-setParam24102, &Mcmd
                   SetParam24102Rq}
testIicpRq         MCMDREQ::={&refMcmd c-mcmd-testIicp, &Mcmd TestIicp}
testConfigRq       MCMDREQ::={&refMcmd c-mcmd-testConfig, &Mcmd
                   TestConfigIICP}

PingIICP::=OCTET STRING (SIZE(0..255))

VCI-info-req::=SEQUENCE{
   medType    ITSatt,
   ciaClass   CIaClass,
   ciClass    CIclass
   }

VCI-update-req::=SEQUENCE (SIZE(0..255)) OF VCI-Info

VCI-Info::=SEQUENCE{
   linkId     Link-ID,
   medType    ITSatt,
   ciaClass   CIaClass,
   ciClass    CIclass,
   status     CIstatus
   }

GetParam24102Rq ::= SEQUENCE OF RefMPARAM

SetParam24102Rq ::= SEQUENCE OF Param24102
```

```
/*
 Simulation of a trigger event in the ITS-S management entity of the SUT
 to perform remote COMMAND access to an MX-SAP in the ITS test system
*/

-- Class for TestIicp
IICPtestRef::=INTEGER{
   c-tiicp-msRcmd     (1),
   c-tiicp-maRcmd     (2),
   c-tiicp-mfRcmd     (3),
   c-tiicp-mnRcmd     (4),
   c-tiicp-miRcmd     (5)
   } (0..255)

IICPTR::=CLASS{
   &cmdRef       IICPtestRef UNIQUE,
   &Cmd
   }

TestIicp::=SEQUENCE{
   commandRef IICPTR.&cmdRef({IicpTestCmds}),
   command    IICPTR.&Cmd({IicpTestCmds}{@commandRef})
   }

IicpTestCmds IICPTR::={msTiicpRcmd | maTiicpRcmd | mfTiicpRcmd |
mnTiicpRcmd | miTiicpRcmd, ...}

-- request to send MS-Command-request to ITS Test System
msTiicpRcmd IICPTR::={&cmdRef c-tiicp-msRcmd, &Cmd MStIICPrcmd}
-- request to send MA-Command-request to ITS Test System
maTiicpRcmd IICPTR::={&cmdRef c-tiicp-maRcmd, &Cmd MAtIICPrcmd}
-- request to send MF-Command-request to ITS Test System
mfTiicpRcmd IICPTR::={&cmdRef c-tiicp-mfRcmd, &Cmd MFtIICPrcmd}
-- request to send MN-Command-request to ITS Test System
mnTiicpRcmd IICPTR::={&cmdRef c-tiicp-mnRcmd, &Cmd MNtIICPrcmd}
-- request to send MI-Command-request to ITS Test System
miTiicpRcmd IICPTR::={&cmdRef c-tiicp-miRcmd, &Cmd MItIICPrcmd}


MStIICPrcmd::=SEQUENCE{
   destination   ITS-scuId,
   cmd           MS-Command-request
   }

MAtIICPrcmd::=SEQUENCE{
   destination   ITS-scuId,
   cmd           MA-Command-request
   }

MFtIICPrcmd::=SEQUENCE{
   destination   ITS-scuId,
   cmd           MF-Command-request
   }

MNtIICPrcmd::=SEQUENCE{
   destination   ITS-scuId,
   cmd           MN-Command-request
   }

MItIICPrcmd::=SEQUENCE{
   destination   ITS-scuId,
   cmd           MF-Command-request
   }

TestConfigIICP::=SEQUENCE{
   location      BIT STRING (SIZE(6)),
   lowerLayers   BIT STRING (SIZE(1)),
   testMode      BIT STRING (SIZE(1))
   }
```

```
-- IIC-Response PDU

IIC-Response::=SEQUENCE{
   sourceITS-scuId         ITS-scuId,
   destinationITS-scuId    ITS-scuId,
   pduCounter              PduCounter,
   responseID              ResponseID, -- value 1
   pduResponse             PduResponse,
   errorStatus             PduErrStatus,
   secRs                   SecRs
   }

ResponseID::=INTEGER{
   response      (1)
   } (0..255)


IICPRES::=CLASS{
   &ref RefIICP UNIQUE,
   &IICPresponse
   }

IICPresponses IICPRES::={aliveRs | mfRcmdRs | mfRreqRs | mnRcmdRs |
mnRreqRs | miRcmdRs | miRreqRs | miRgetRs | miRsetRs | mCmdRs |
maRcmdRs | maRreqRs | msRcmdRs | msRreqRs /* | simAutCmdRs |
simAutReqRs */ | simFutCmdRs | simFutReqRs | simNutCmdRs |
simNutReqRs | simIutCmdRs | simIutReqRs /* | simAltCmdRs |
simAltReqRs */ | simFltCmdRs | simFltReqRs | simNltCmdRs |
simNltReqRs | simUtInRs | simUtOutRs, ...}

PduResponse::=SEQUENCE{
   resRef    IICPRES.&ref({IICPresponses}), -- data type identifier
   iicpRes   IICPRES.&IICPresponse({IICPresponses}{@resRef})
   }

-- for operational usage
aliveRs       IICPRES::={&ref c-iic-alive, &IICPresponse ITS-SCUalive}
mfRcmdRs      IICPRES::={&ref c-iic-mfRcmd, &IICPresponse MF-Command-
              confirm}
mfRreqRs      IICPRES::={&ref c-iic-mfRreq, &IICPresponse MF-Request-
              confirm}
mnRcmdRs      IICPRES::={&ref c-iic-mnRcmd, &IICPresponse MN-Command-
              confirm}
mnRreqRs      IICPRES::={&ref c-iic-mnRreq, &IICPresponse MN-Request-
              confirm}
miRcmdRs      IICPRES::={&ref c-iic-miRcmd, &IICPresponse MI-Command-
              confirm}
miRreqRs      IICPRES::={&ref c-iic-miRreq, &IICPresponse MI-Request-
              confirm}
miRgetRs      IICPRES::={&ref c-iic-miRget, &IICPresponse MI-Get-
              confirm}
miRsetRs      IICPRES::={&ref c-iic-miRset, &IICPresponse MI-Set-
              confirm}
mCmdRs        IICPRES::={&ref c-iic-mCmd, &IICPresponse McmdRs}
maRcmdRs      IICPRES::={&ref c-iic-maRcmd, &IICPresponse MA-Command-
              confirm}
maRreqRs      IICPRES::={&ref c-iic-maRreq, &IICPresponse MA-Request-
               confirm}
msRcmdRs      IICPRES::={&ref c-iic-msRcmd, &IICPresponse MS-Command-
              confirm}
msRreqRs      IICPRES::={&ref c-iic-msRreq, &IICPresponse MS-Request-
              confirm}
-- for test usage only
-- simAutCmdRs   IICPRES::={&ref c-iic-simAutCmd, &IICPresponse
                 NullType}
-- simAutReqRs   IICPRES::={&ref c-iic-simAutReq, &IICPresponse
                 NullType}
simFutCmdRs   IICPRES::={&ref c-iic-simFutCmd, &IICPresponse NullType}
simFutReqRs   IICPRES::={&ref c-iic-simFutReq, &IICPresponse NullType}
simNutCmdRs   IICPRES::={&ref c-iic-simNutCmd, &IICPresponse NullType}
simNutReqRs   IICPRES::={&ref c-iic-simNutReq, &IICPresponse NullType}
```

```
simIutCmdRs    IICPRES::={&ref c-iic-simIutCmd, &IICPresponse NullType}
simIutReqRs    IICPRES::={&ref c-iic-simIutReq, &IICPresponse NullType}
-- simAltCmdRs    IICPRES::={&ref c-iic-simAltCmd, &IICPresponse
                  NullType}
-- simAltReqRs    IICPRES::={&ref c-iic-simAltReq, &IICPresponse
                  NullType}
simFltCmdRs    IICPRES::={&ref c-iic-simFltCmd, &IICPresponse NullType}
simFltReqRs    IICPRES::={&ref c-iic-simFltReq, &IICPresponse NullType}
simNltReqRs    IICPRES::={&ref c-iic-simNltReq, &IICPresponse NullType}
simNltCmdRs    IICPRES::={&ref c-iic-simNltCmd, &IICPresponse NullType}
simUtInRs      IICPRES::={&ref c-iic-simUtIn, &IICPresponse NullType}
simUtOutRs     IICPRES::={&ref c-iic-simUtOut, &IICPresponse NullType}

MCMDRES::=MCMD

McmdRess MCMDRES::={pingRs | vCIinfoRs | vCIupdateRs | getParam24102Rs |
setParam24102Rs | testIicpRs | testConfigRs, ...}

McmdRs::=SEQUENCE{
   mCmdRef    MCMDRES.&refMcmd({McmdRess}),
   mcmd       MCMDRES.&Mcmd ({McmdRess}{@mCmdRef})
   }

pingRs             MCMDRES::={&refMcmd c-mcmd-ping, &Mcmd PingIICP}
vCIinfoRs          MCMDRES::={&refMcmd c-mcmd-vCIinfo,
                   &Mcmd VCI-info-res}
vCIupdateRs        MCMDRES::={&refMcmd c-mcmd-vCIupdate, &Mcmd NullType}
getParam24102Rs    MCMDRES::={&refMcmd c-mcmd-getParam24102, &Mcmd
                   GetParam24102Rs}
setParam24102Rs    MCMDRES::={&refMcmd c-mcmd-setParam24102, &Mcmd
                   SetParam24102Rs}
testIicpRs         MCMDRES::={&refMcmd c-mcmd-testIicp, &Mcmd NullType}
testConfigRs      MCMDRES::={&refMcmd c-mcmd-testConfig, &Mcmd NullType}

PduErrStatus::=INTEGER{
   success                (0),
   pduUnknown             (1),
   duplicateITS-scuId     (2),
   invalidAliveMessage    (3),
   invalidITSscuType      (4),
-- reserved for future usage  (5) .. (127)
-- reserved for implementation-specific usage (128) .. (254)
   unspecFailure          (255)
   } (0..255)

VCI-info-res::=SEQUENCE (SIZE(0..255)) OF VCI-Info

GetParam24102Rs ::= SEQUENCE{
   paramOK       Param24102OK,
   paramNOK      Param24102NOK
   }

Param24102OK ::= SEQUENCE OF Param24102

Param24102NOK ::= SEQUENCE OF Param24102Error

Param24102Error ::= SEQUENCE{
   paramNo       RefMPARAM,
   error         ErrStatus
   }

SetParam24102Rs ::= SEQUENCE OF Param24102Error

-- IIC-Request Security

RefSECIICP ::= INTEGER {
   c-secNoSecurity    (0),
   c-secOctetString   (1)
   } (0..255)

SECIICP::=CLASS{
```

```
   &ref RefSECIICP, -- security type identifier
   &SecIICP
   }

SecIICPs SECIICP::={noSecurity | octString, ...}

noSecurity      SECIICP::={&ref c-secNoSecurity, &SecIICP NullType}
octString       SECIICP::={&ref c-secOctetString, &SecIICP IICPsecurityContainer}

IICPsecurityContainer::=OCTET STRING

SecRq::=SEQUENCE{
   secRqRef     SECIICP.&ref({SecIICPs}),
   secRq        SECIICP.&SecIICP({SecIICPs}{@secRqRef})
   }

SecRs::=SEQUENCE{
   secRsRef     SECIICP.&ref({SecIICPs}),
   secRs        SECIICP.&SecIICP({SecIICPs}{@secRsRef})
   }

-- MF-SAP --
-- MF-COMMANDs --

IICrequestTX::=IIC-Request

IICresponseTX::=IIC-Response

-- MF-REQUESTs --

IICrequestRX::=IIC-Request

IICresponseRX::=IIC-Response

-- General types --

PduCounter::=INTEGER(0..65535)

-- Values


version INTEGER(0..255)::=2 -- version of this ASN.1 module

/*
   The ASN.1 specification has been checked for conformance to the ASN.1
   standards by OSS ASN.1 Syntax Checker, and by OSS ASN-1STEP
*/

END
```

Annex B
(normative)

IIC PDUs

B.1 List of PDUs

Table B.1 presents an overview of all IIC PDUs. Further details are specified in the next sub-clauses of this annex. As presented in Table 1, IIC request and response PDUs are defined. A response PDU may contain an empty Data field. The column "PDU-ID" indicates the reference number of the PDU type. Further PDU types may apply as being registered without the need to be listed in this document.

Table B.1 — IIC PDUs

Type of PDU	PDU-ID	Comment
ITS-SCUalive	`c-iic-alive` = 0	Used to assign, maintain and delete unique ITS-SCU-ID values in a station.
MF-rcmd	`c-iic-mfRcmd` = 1	A management command MF-COMMAND issued by the ITS station management entity of the local ITS-SCU, to be forwarded to the MF-SAP of one or several remote ITS-SCUs.
MF-rreq	`c-iic-mfRcmd` = 2	A command MF-REQUEST issued by the local facilities layer, to be forwarded to the ITS station management entity in one or several remote ITS-SCUs.
MN-rcmd	`c-iic-mnRcmd` = 3	A management command MN-COMMAND issued by the ITS station management entity of the local ITS-SCU, to be forwarded to the MN-SAP of one or several remote ITS-SCUs.
MN-rreq	`c-iic-mnRreq` = 4	A command MN-REQUEST issued by the local networking and transport layer, to be forwarded to the ITS station management entity in one or several remote ITS-SCUs.
MI-rcmd	`c-iic-miRcmd` = 5	A management command MI-COMMAND issued by the ITS station management entity of the local ITS-SCU, to be forwarded to the MI-SAP of a remote ITS-SCUs.
MI-rreq	`c-iic-miRreq` = 6	A command MI-REQUEST issued by the local access layer, to be forwarded to the ITS station management entity in one or several remote ITS-SCUs.
MI-rget	`c-iic-miRget` = 7	A command MI-GETPARAM issued by the ITS station management entity, to be forwarded to the MI-SAP of a remote ITS-SCU.
MI-rset	`c-iic-miRset` = 8	A command MI-SETPARAM issued by the ITS station management entity, to be forwarded to the MI-SAP of a remote ITS-SCU.
Mcmd	`c-iic-mCmd` = 9	Request executing of a management command in the ITS-S management entity of a remote ITS-SCU. Details see in the rows below.
Mcmd.Ping	`RefMcmd`: `c-mcmd-ping` = 0	Octet string with random data to be replied.

Table B.1 *(continued)*

Type of PDU	PDU-ID	Comment
Mcmd.VCI-info	`RefMcmd`: `c-mcmd-vCIinfo` = 1	Request to all ITS-SCUs containing a router, to report about existing VCIs. Information to be stored in VCI list.
Mcmd.VCI-update	`RefMcmd`: `c-mcmd-vCIupdate` = 2	Information on change of VCI information to be stored in VCI list. Broadcasted to all ITS-SCUs.
Mcmd.GET-Param24102	`RefMcmd`: `c-mcmd-getParam24102` = 3	Retrieves the value of a management parameter Param24102 specified in ISO 24102-1 from another ITS-SCU.
Mcmd.SET-Param24102	`RefMcmd`: `c-mcmd-setParam24102` = 4	Set the value of a management parameter Param24102 specified in ISO 24102-1 from another ITS-SCU.
Mcmd.TestIicp	`RefMcmd`: `c-mcmd-testIicp` = 254	Optional feature, used for test purposes only. See ISO/TS 20026[4] Simulation of a trigger event in the ITS-S management entity of the SUT to perform remote COMMAND access to an MX-SAP in the ITS test system.
Mcmd.TestConfigIICP	`RefMcmd`: `c-mcmd-testConfig` = 255	Optional feature, used for test purposes only. See ISO/TS 20026[4] Used to program the test mode in an ITS-SCU.
MA-rcmd	`c-iic-maRcmd` = 10	A management command MA-COMMAND issued by the ITS station management entity of the local ITS-SCU, to be forwarded to the MA-SAP of one or several remote ITS-SCUs.
MA-rreq	`c-iic-maRreq` = 11	A command MA-REQUEST issued by the local facilities layer, to be forwarded to the ITS station management entity in one or several remote ITS-SCUs.
MS-rcmd	`c-iic-msRcmd` = 12	A management command MS-COMMAND issued by the ITS station management entity of the local ITS-SCU, to be forwarded to the MS-SAP of one or several remote ITS-SCUs.
MS-rreq	`c-iic-msRreq` = 13	A command MS-REQUEST issued by the local ITS-S security entity, to be forwarded to the ITS station management entity in one or several remote ITS-SCUs.
—	14 to 235	To be assigned by the registry
The following PDU types provide optional features that are used for test purposes only.		
—	236 to 239	Reserved for potential future introduction of SimSUTcmd, SimSUTreq, SimSLTcmd, SimSLTreq
SimUtIn	`c-iic-simUtIn` = 240	See ISO/TS 20026[4] and ISO 24102-3.
SimUtOut	`c-iic-simUtOut` = 241	See ISO/TS 20026[4] and ISO 24102-3.
SimAUTcmd	`c-iic-simAutCmd` = 242	So far not supported in ISO/TS 20026[4] and ISO 24102-3.
SimAUTreq	`c-iic-simAutReq` = 243	So far not supported in ISO/TS 20026[4] and ISO 24102-3.
SimFUTcmd	`c-iic-simFutCmd` = 244	See ISO/TS 20026[4] and ISO 24102-3.
SimFUTreq	`c-iic-simFutReq` = 245	See ISO/TS 20026[4] and ISO 24102-3.
SimNUTcmd	`c-iic-simNutCmd` = 246	See ISO/TS 20026[4] and ISO 24102-3.

Table B.1 *(continued)*

Type of PDU	PDU-ID	Comment
SimNUTreq	`c-iic-simNutReq` = 247	See ISO/TS 20026[4] and ISO 24102-3.
SimIUTcmd	`c-iic-simIutCmd` = 248	See ISO/TS 20026[4] and ISO 24102-3.
SimIUTreq	`c-iic-simIutReq` = 249	See ISO/TS 20026[4] and ISO 24102-3.
SimALTcmd	`c-iic-simAltCmd` = 250	See ISO/TS 20026[4] and ISO 24102-3.
SimSLTreq	`c-iic-simAltReq` = 251	See ISO/TS 20026[4] and ISO 24102-3.
SimFLTcmd	`c-iic-simFltCmd` = 252	See ISO/TS 20026[4] and ISO 24102-3.
SimFLTreq	`c-iic-simFltReq` = 253	See ISO/TS 20026[4] and ISO 24102-3.
SimNLTcmd	`c-iic-simNltCmd` = 254	See ISO/TS 20026[4] and ISO 24102-3.
SimNLTreq	`c-iic-simNltReq` = 255	See ISO/TS 20026[4] and ISO 24102-3.

B.2 PDU details

B.2.1 ASN.1

ASN.1 details of all PDUs shall be as presented in Annex A.

B.2.2 ITS-SCUalive

Table B.2 shows details of the "Data" element of ASN.1 type `ITS-SCUalive` in the IIC-Request PDU.

Table B.2 — ITS-SCUalive request PDU

ASN.1 component	Description
`message`	Indicates type of alive message: 0: "alive" 1: "delete" 255: "new"
`its-scuType`	Indicates role of ITS-SCU: 0: unknown 1: ITS-S host 2: ITS-S router 3: ITS test system 4: Test CI unit 5: ITS-S host and router

This request shall always be transmitted to all ITS-SCUs.

Table B.3 shows details of the "Data" element of ASN.1 type `ITS-SCUalive` in the IIC-Response PDU.

Table B.3 — ITS-SCUalive response PDU

ASN.1 component	Description
`message`	Same as in related request
`its-scuType`	Same as in related request

With "Alive Message" = "delete" in the IIC-Request PDU, no IIC-Response PDU shall be transmitted.

B.2.3 Remote SAP access

The "Data" element in the IIC-Request PDUs contains the SAP service primitives with the following ASN.1 types:

— MA-rcmd: ASN.1 type `MA-Command-request`;

— MF-rcmd: ASN.1 type `MF-Command-request`;

— MN-rcmd: ASN.1 type `MN-Command-request`;

— MI-rcmd: ASN.1 type `MI-Command-request`;

— MS-rcmd: ASN.1 type `MS-Command-request`;

— MA-rreq: ASN.1 type `MA-Request-request`;

— MF-rreq: ASN.1 type `MF-Request-request`;

— MN-rreq: ASN.1 type `MN-Request-request`;

— MI-rreq: ASN.1 type `MI-Request-request`;

— MI-rget: ASN.1 type `MI-Get-request`;

— MI-rset: ASN.1 type `MI-Set-request`;

— MS-rreq: ASN.1 type `MS-Request-request`.

specified in ISO 24102-3, see Annex A.

The "Data" element in the IIC-Response PDUs contains the SAP service primitives with the following ASN.1 types:

— MA-rcmd: ASN.1 type `MA-Command-confirm`;

— MF-rcmd: ASN.1 type `MF-Command-confirm`;

— MN-rcmd: ASN.1 type `MN-Command-confirm`;

— MI-rcmd: ASN.1 type `MI-Command-confirm`;

— MS-rcmd: ASN.1 type `MS-Command-confirm`;

— MA-rreq: ASN.1 type `MA-Request-confirm`;

— MF-rreq: ASN.1 type `MF-Request-confirm`;

— MN-rreq: ASN.1 type `MN-Request-confirm`;

— MI-rreq: ASN.1 type `MI-Request-confirm`;

— MI-rget: ASN.1 type `MI-Get-confirm`;

— MI-rset: ASN.1 type `MI-Set-confirm`;

— MS-rreq: ASN.1 type `MS-Request-confirm`.

specified in ISO 24102-3, see Annex A.

Before the receiving IICM forwards a command contained in an IIC-Request PDU to the appropriate layer it shall temporarily store "CommandRef", and shall replace "CommandRef" in the forwarded SAP service primitive by its locally generated value. The locally stored value of "CommandRef" shall be used in the IIC-Response PDU related to this command.

Any kind of remote access shall be controlled by the IICM which receives the request, i.e. allowing rejecting a request in case it is not acceptable. Details are outside the scope of this document.

B.2.4 Mcmd

B.2.4.1 General

Mcmd provides a set of management command to be executed in the ITS-S management entity of a remote ITS-SCU. The following management commands are identified by the reference number of ASN.1 type `RefMcmd`:

— Ping command: `c-mcmd-ping` = 0;

— VCI-info command: `c-mcmd-vCIinfo` = 1;

— VCI-update command: `c-mcmd-vCIupdate` = 2;

— GetParam24102 command: `c-mcmd-getParam24102` = 3;

— SetParam24102 command: `c-mcmd-setParam24102` = 4;

— TestIICP command: `c-mcmd-testIicp` = 254;

— TestConfig command: `c-mcmd-testConfig` = 255.

B.2.4.2 Ping command

Table B.4 shows details of the "Data" element of ASN.1 type `McmdRq` in the IIC-Request PDU for the Ping command of ASN.1 type `PingIICP`.

Table B.4 — Ping request PDU

ASN.1 type	Description
`PingIICP`	Octet string with size of zero through 255 octets.

Upon reception of this request, the IICM shall reply the received octet string in the related IIC-Response PDU.

B.2.4.3 VCI-info

Table B.5 shows details of the "Data" element of ASN.1 type `McmdRq` in the IIC-Request PDU for the VCI-Info command of ASN.1 type `VCI-info-req`.

Table B.5 — VCI-info request PDU

ASN.1 component	Description
`medType`	Indicates requested type of medium as specified in ISO 21218.
`ciaClass`	Indicates requested CI access class as specified in ISO 21218.
`ciclass`	Indicates requested CI class as specified in ISO 21218.

Upon reception of this request, the IICM shall check the required properties of existing CIs/VCIs, and shall report the information in the IIC-Response PDU related to this command. The three requirements shall simultaneously be fulfilled for all information reported in the related IIC-Response.

Table B.6 shows details of the "Data" element of ASN.1 type `McmdRs` in the IIC-Response PDU for the VCI-Info response of ASN.1 type `VCI-info-res` reported for every CI/VCI. `VCI-info-res` is a sequence of `VCI-Info`.

Table B.6 — VCI-info response PDU

ASN.1 component of VCI-Info	Description
linkId	As specified in ISO 21218.
medType	As specified in ISO 21218.
ciaClass	As specified in ISO 21218.
ciclass	As specified in ISO 21218.
status	As specified in ISO 21218.

B.2.4.4 VCI-update

Table B.7 shows details of the "Data" element of ASN.1 type McmdRq in the IIC-Request PDU for the VCI-update command of ASN.1 type VCI-update-req. This PDU shall be sent in broadcast mode to all ITS-SCUs. VCI-update-req is a sequence of VCI-Info.

Table B.7 — VCI-updatse request PDU

ASN.1 component of VCI-Info	Description
linkId	As specified in ISO 21218.
medType	As specified in ISO 21218.
ciaClass	As specified in ISO 21218.
ciclass	As specified in ISO 21218.
status	As specified in ISO 21218.

This message shall not be acknowledged.

B.2.4.5 GET-Param24102

Table B.8 shows details of the "Data" element of ASN.1 type McmdRq in the IIC-Request PDU for the Get-Param24102 command of ASN.1 type GetParam24102Rq.

Table B.8 — GET-Param24102 request PDU

ASN.1 type	Description
GetParam24102Rq	Sequence of RefMPARAM

This request shall be transmitted only to a single ITS-SCU.

Table B.9 shows details of the "Data" element of ASN.1 type McmdRs in the IIC-Response PDU for the Get-Param24102 response of ASN.1 type GetParam24102Rs.

Table B.9 — GET-Param24102 response PDU

ASN.1 component	Description
paramOK	Sequence of successfully retrieved parameter values
paramNOK	Sequence of error codes related to requested parameter values that could not be retrieved.

B.2.4.6 SET-Param24102

Table B.10 shows details of the "Data" element of ASN.1 type McmdRq in the IIC-Request PDU for the Set-Param24102 command of ASN.1 type SetParam24102Rq.

Table B.10 — SET-Param24102 request PDU

ASN.1 type	Description
SetParam24102Rq	Sequence of Param24102

This request shall be transmitted only to a single ITS-SCUs.

Table B.11 shows details of the "Data" element of ASN.1 type McmdRs in the IIC-Response PDU for the Set-Parameter24102 response of ASN.1 type SetParam24102Rs.

Table B.11 — SET-Param24102 response PDU

ASN.1 type	Description
SetParam24102Rs	Sequence of Param24102Error indicating success or failure of setting of a parameter value.

B.2.4.7 TestIicp

Table B.12 shows details of the "Data" element of ASN.1 type McmdRq in the IIC-Request PDU for the TestIICP command of ASN.1 type TestIicp.

Table B.12 — TestIicp request PDU

ASN.1 component	Description
commandRef	Integer number selecting the command; see ASN.1 type IICPtestRef
command	For available commands see the ASN.1 type IicpTestCmds.

This request is used only for conformance test purposes. It shall be ignored by a receiving ITS-SCU which is not in test mode, see B.2.4.8. It shall not be transmitted by an ITS-SCU which is not performing the role of an ITS test system.

This message shall be acknowledged with an empty Data field.

Details of usage of this PDU type are specified in ISO/TS 20026[4].

B.2.4.8 TestConfigIICP

Table B.13 shows details of the "Data" element of ASN.1 type McmdRq in the IIC-Request PDU for the TestConfigIICP command of ASN.1 type TestConfigIICP.

Table B.13 — TestConfigIicp request PDU

ASN.1 component	Description
location	String of six bits indicating location of the implementation under test
lowerLayers	Binary flag indicating whether lower layers in the system under test are used to access the implementation under test
testMode	Binary flag to switch on and off the test mode. This remote activation and deactivation of the test mode in the system under test is only possible after authorization in the system under test by a manufacturer-specific means.

This request is used only for conformance test purposes. It shall be ignored by a receiving ITS-SCU which is not in test mode.

This message shall be acknowledged with an empty Data field.

Details of usage of this PDU type are specified in ISO/TS 20026[4].

B.2.4.9 Upper and lower tester access simulation

The functionality of the PDU types with PDU-ID in the range from 240 to 255 are specified in ISO/TS 20026[4] and in ISO 24102-3.

These requests are used only for conformance test purposes. These shall be ignored by a receiving ITS-SCU which is not in test mode. These shall not be transmitted by an ITS-SCU which is not in test mode.

These messages shall be acknowledged with an empty Data field.

Annex C
(normative)

Implementation conformance statement (ICS) proforma

Users of this document may

— freely reproduce the ICS proforma in this annex so that it can be used for its intended purposes and

— may further publish the completed ICS.

C.1 Guidance for completing the ICS proforma

C.1.1 Purposes and structure

The purpose of this Implementation Conformance Statement (ICS) proforma is to provide a mechanism whereby a supplier of an implementation of the requirements defined in this document may provide information about the implementation in a standardized manner.

The ICS proforma is subdivided into clauses for the following categories of information:

— guidance for completing the ICS proforma;

— identification of the implementation;

— identification of the implementation;

— global statement of conformance.

C.1.2 Abbreviations and conventions

The ICS proforma contained in this annex is comprised of information in tabular form in accordance with the guidelines presented in ISO/IEC 9646-7[6].

Item column	The item column contains a number which identifies the item in the table.
Item description column	The item description column describes in free text each respective item (e.g. parameters).
Status column	The notations defined in ISO/IEC 9646-7[6] are used for the status column: m mandatory – the capability is required to be supported. o optional – the capability may be supported or not. n/a not applicable – in the given context, it is impossible to use the capability. x prohibited (excluded) – there is a requirement not to use this capability in the given context. o.i qualified optional – for mutually exclusive or selectable options from a set. "i" is an integer which identifies an unique group of related optional items and the logic of their selection which is defined immediately following the table.

	ci conditional – the requirement on the capability ("m", "o", "x" or "n/a") depends on the support of other optional or conditional items. "i" is a string containing the respective Table number, followed by a dash, followed by a sequential number identifying a unique conditional status expression which is defined immediately following the respective Table. r as specified in the related referenced standard of the CI access technology.
Reference column	The reference column makes reference to this document, except where explicitly stated otherwise.
Support column	The support column shall be filled in by the supplier of the implementation. The following common notations, defined in ISO/IEC 9646-7[6], are used for the support column: Y or y supported by the implementation. N or n not supported by the implementation. N/A, n/a, or no answer required (allowed only if the status is n/a, directly or after evaluation of a conditional status).
Values allowed column	The values allowed column contains the type, the list, the range, or the length of values allowed. The following notations are used: — range of values: <min value> .. <max value> — list of values: <value1>, <value2>, ..., <valueN> — list of named values: <name1>(<val1>), <name2>(<val2>), ..., <nameN>(<valN>) — length: size (<min size> .. <max size>)
Values supported column	The values supported column shall be filled in by the supplier of the implementation. In this column, the values or the ranges of values supported by the implementation shall be indicated.
References to items	For each possible item answer (answer in the support column) within the ICS proforma a unique reference exists, used, for example, in the conditional expressions. It is defined as the table identifier, followed by a solidus character "/", followed by the item number in the table. If there is more than one support column in a table, the columns are discriminated by letters (a, b, etc.), respectively.
Prerequisite line	A prerequisite line takes the form: Prerequisite: <predicate>. A prerequisite line after a clause or table title indicates that the whole clause or the whole table is not required to be completed if the predicate is FALSE.

C.1.3 Instructions for completing the ICS proforma

The supplier of the implementation shall complete the ICS proforma. In particular, an explicit answer shall be entered using the notation described in C.1.2.

C.2 Identification of the Implementation

Identification of the Implementation Under Test (IUT) and the system in which it resides, i.e. the System Under Test (SUT), shall be filled in so as to provide as much detail as possible regarding version numbers and configuration options.

The product supplier information and client information shall both be filled in if they are different.

A person who can answer queries regarding information supplied in the ICS shall be named as the contact person.

NOTE The ICS template to be used starts on the next page.

C.2.1 Date of the statement

_____

C.2.2 Identification of IUT

Name of Implementation under Test (IUT):

_____

_____

Version of IUT:

_____

C.2.3 Identification of SUT

Name of System under Test (SUT):

_____

_____

Hardware configuration:

_____

_____

_____

_____

Operating system:

_____

_____

C.2.4 Product supplier

Name:

_____

Address:

_____

_____

_____

Telephone number:

_____

Facsimile number:

Email address:

Additional information:

C.2.5 Client

Name:

Address:

Telephone number:

Facsimile number:

Email address:

Additional information:

C.2.6 ICS contact person

Name:

Telephone number:

Facsimile number:

_____

Email address:

_____

Additional information:

_____

_____

_____

_____

C.3 Implementation specification

This ICS proforma applies to the International Standard ISO 24102-4 (this document).

C.4 Global statement of conformance

Are all mandatory capabilities implemented? (Yes/No) _____

NOTE Answering "No" to this question indicates non-conformance to the implementation specification. Non supported mandatory capabilities are to be identified in the ICS, with an explanation of why the implementation is non-conforming, on pages attached to the ICS proforma.

C.5 Detailed conformance statements

C.5.1 Design choices

Table C.1 — Design choices

Item	Supported design choices	Reference	Status	Support
1	Applying IICP as specified in ISO/TS 20026[4].	—	o	
2	Stand-alone ITS-S host	Clause 6	o.1	
3	Stand-alone ITS-S router	Clause 6	o.1	
4	Combined ITS-S host/router	Clause 6	o.1	
5	Implementation of management procedure is testable	9.1, 9.2	o.2	
6	Implementation of management procedure is not testable	9.1, 9.2	o.2	

o.1: It is mandatory to support at least one of these items.

o.2: It is mandatory to support at least one of these items.