BS ISO 24102-3:2013

BSI Standards Publication

Intelligent transport systems — Communications access for land mobiles (CALM) — ITS station management

Part 3: Service access points

National foreword

This British Standard is the UK implementation of ISO 24102-3:2013. Together with BS ISO 24102-1:2013, BS ISO 24102-2, BS ISO 24102-4:2013 and BS ISO 24102-5:2013, it supersedes BS ISO 24102:2010, which will be withdrawn upon publication of BS ISO 24102-2 of this series.

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 secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

ISBN 978 0 580 79274 8

ICS 03.220.01; 35.240.60

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 July 2013.

Amendments issued since publication

Date	Text affected

BS ISO 24102-3:2013

INTERNATIONAL STANDARD

ISO 24102-3

First edition
2013-07-01

Intelligent transport systems — Communications access for land mobiles (CALM) — ITS station management —

Part 3: Service access points

Systèmes intelligents de transport — Accès aux communications des services mobiles terrestres (CALM) — Gestion des stations ITS —

Partie 3: Points d'accès au service

Reference number
ISO 24102-3:2013(E)

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. 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. 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.

The committee responsible for this document is ISO/TC 204, *Intelligent transport systems*.

ISO 24102 consists of the following parts, under the general title *Intelligent transport systems — Communications access for land mobiles (CALM) — ITS station management*:

— *Part 1: Local management*

— *Part 3: Service access points*

— *Part 4: ITS station-internal management communications*

— *Part 5: Fast service advertisement protocol (FSAP)*

The following parts are under preparation:

— *Part 2: Remote management*

— *Part 6: Path and flow management*

Introduction

This International Standard is part of a family of International Standards for communications access for land mobiles (CALM). An introduction to the whole set of International Standards is provided in ISO 21217.

This part of ISO 24102 is part 3 of a multipart International Standard which determines the intelligent transport systems (ITS) station management service access points.

The ITS station management entity provides functionality related to the management of communication protocol layers and the security entity presented in the ITS station reference architecture specified in ISO 21217 and presented in Figure 1, and in line with the general ITS architecture specified in ISO 21217.

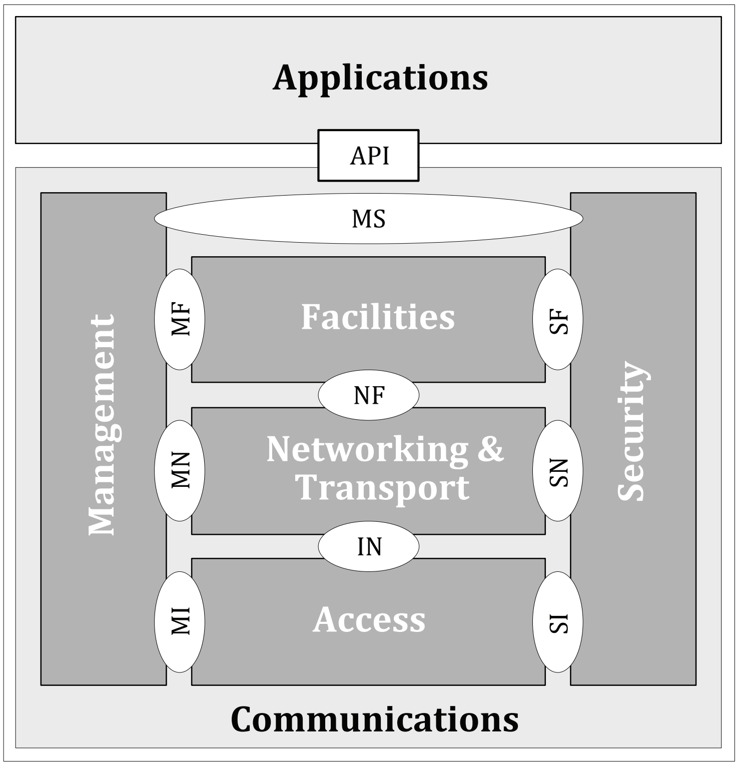

Figure 1 — ITS station reference architecture with named interfaces

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

Intelligent transport systems — Communications access for land mobiles (CALM) — ITS station management —

Part 3: **Service access points**

1 Scope

This part of ISO 24102 specifies the management service access points, i.e. the service access points of

- the interfaces between the ITS-S management entity and
  - the ITS-S access layer (MI-SAP),
  - the ITS-S networking and transport layer (MN-SAP),
  - the ITS-S facilities layer (MF-SAP),
- the interfaces between the ITS-S security entity and
  - the ITS-S access layer,
  - the ITS-S networking and transport layer,
  - the ITS-S facilities layer,
- the interface between the ITS-S management entity and the ITS-S security entity (MS-SAP).

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. 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) — Part 2*

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

ISO 21218, *Intelligent transport systems — Communications access for land mobiles (CALM) — Access technology support*

ISO 24102-1, *Intelligent transport systems — Communications access for land mobiles (CALM) — ITS station management — Part 1: Local management*

ISO 24102-4, *Intelligent transport systems — Communications access for land mobiles (CALM) — ITS station management — Part 4: Station-internal management communications*

ISO 24102-5, *Intelligent transport systems — Communications access for land mobiles (CALM) — ITS station management — Part 5: Fast service advertisement protocol (FSAP)*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 21217 apply.

4 Abbreviated terms

For the purposes of this document, the abbreviated terms given in ISO 21217 and the following apply.

MF-COMMAND command issued by the ITS-S management entity and sent to the ITS-S facilities layer via the MF-SAP

MF-REQUEST command issued by the ITS-S facilities layer and sent to the ITS-S management entity via the MF-SAP

MI-COMMAND command issued by the ITS-S management entity and sent to the ITS-S access layer via the MI-SAP

MI-GET command issued by the ITS-S management entity in order to get the value of one or several CI parameters; sent via the MI-SAP to the ITS-S access layer

MI-REQUEST command issued by the ITS-S access layer and sent to the ITS-S management entity via the MI-SAP

MI-SET command issued by the ITS-S management entity in order to set the value of one or several CI parameters; sent via the MI-SAP to the ITS-S access layer

MN-COMMAND command issued by the ITS-S management entity and sent to the ITS-S networking and transport layer via the MN-SAP

MN-REQUEST command issued by the ITS-S networking and transport layer and sent to the ITS-S management entity via the MN-SAP

MS-COMMAND command issued by the ITS-S management entity and sent to the ITS-S security entity via the MS-SAP

MS-REQUEST command issued by the ITS-S security entity and sent to the ITS-S management entity via the MS-SAP

SF-COMMAND command issued by the ITS-S security entity and sent to the ITS-S facilities layer via the SF-SAP

SF-REQUEST command issued by the ITS-S facilities layer and sent to the ITS-S security entity via the SF-SAP

SI-COMMAND command issued by the ITS-S security entity and sent to the ITS-S access layer via the SI-SAP

SI-REQUEST command issued by the ITS-S access layer and sent to the ITS-S security entity via the SI-SAP

SN-COMMAND command issued by the ITS-S security entity and sent to the ITS-S networking and transport layer via the SN-SAP

SN-REQUEST command issued by the ITS-S networking and transport layer and sent to the ITS-S security entity via the SN-SAP

5 Requirements

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

1) General management functionality specified in ISO 24102-1.

2) The functionality of remote ITS station management specified in ISO 24102-2.

3) The functionality of management/security service access points specified this part of ISO 24102.

4) The functionality of ITS station-internal management communications specified in ISO 24102-4.

5) The functionality of the "Fast Service Advertisement Protocol" (FSAP) specified in ISO 24102-5.

The functionality of the service access points

— MI-SAP, MN-SAP, MF-SAP,

— SI-SAP, SN-SAP, SF-SAP,

— MS-SAP,

shall be as specified in this part of ISO 24102.

NOTE 1 SAPs can be non-exposed interfaces. Thus, service primitives of SAPs are not subject to conformance tests.

NOTE 2 The MA-interface and SA-interface presented in Figure 1 are not specified in this part of ISO 24102, although these are recognized as part of the ITS station reference architecture specified in ISO 21217. All interfaces towards the ITS-S applications are considered to be APIs rather than SAPs.

Means to secure the access to management functionality need to be specified within the global context of CALM security. Details are outside the scope of this part of ISO 24102.

Detailed mandatory requirements are specified in the following clauses of this part of ISO 24102:

— Clause 6 specifies the MI-SAP.

— Clause 7 specifies the MN-SAP.

— Clause 8 specifies the MF-SAP.

— Clause 9 specifies the SI-SAP.

— Clause 10 specifies the SN-SAP.

— Clause 11 specifies the SF-SAP.

— Clause 12 specifies the MS-SAP.

— Annexes provide further mandatory requirements.

6 MI-SAP

6.1 General

Basically, there are the following different types of services:

a) a service that allows the ITS-S management entity to set I-Parameter values in a "(Virtual) Communication Interface" (CI, VCI), see ISO 21218;

b) a service that allows the ITS-S management entity to read I-Parameter values from a (virtual) CI;

c) a service that allows a "Management Adaptation Entity" (MAE), see ISO 21218, to notify events to the ITS-S management, including changes in (virtual) CI I-Parameter values;

d) a service that allows the ITS-S management entity to send a command to an MAE;

e) a service that allows an MAE to send a command to the ITS-S management entity;

f) a service for transmission requests of a management data packet;

g) a service for reception notification of a management data packet.

Setting of I-Parameter values shall be built on the service MI-SET.

Reading of I-Parameter values shall be built on the service MI-GET.

Issuing of commands by the ITS-S management entity shall be built on the service MI-COMMAND.

Issuing of commands by a CI or VCI shall be built on the service MI-REQUEST.

Transmission request and reception notification of management data packets shall be built on the services MI-COMMAND and MI-REQUEST.

Event notifications shall be built on the service MI-REQUEST.

Every request and confirm service primitive shall be identified by a CommandRef parameter. A value unique within the available number space shall be assigned to the request service primitive. The related confirm service primitive shall use the same number as provided in the related request service primitive.

Management communications with the access layer in other ITS-SCUs shall be with Inter-ITS-SCU communication PDUs "MI-rset", "MI-rget", "MI-rcmd", and "MI-rreq" as specified in ISO 24102-4.

Details of the service primitives shall be as specified in Annex A.

6.2 MI-SET

6.2.1 MI-SET.request

The service primitive MI-SET.request allows setting of I-Parameters by the ITS-S management.

The parameters of MI-SET.request are as follows:

```
MI-SET.request              (
                            Link-ID,
                            CommandRef,
                            Sequence of I-Param
                            )
```

Table 1 — MI-SET.request parameter description

Name	Description
Link-ID	Unique identifier of a VCI/CI.
CommandRef	Unique cyclic reference number of command.
-	Number of subsequent elements I-Param.
I-Param.No	See ISO 21218.
I-Param.Value	See ISO 21218.

An MI-SET.request is generated by the ITS-S management entity when I-Parameters specified in ISO 21218 shall be set.

On receipt of MI-SET.request, the selected parameters shall be set if applicable.

6.2.2 MI-SET.confirm

The service primitive MI-SET.confirm reports the result of a previous MI-SET.request.

The parameters of MI-SET.confirm are as follows:

```
MI-SET.confirm          (
                        Link-ID,
                        CommandRef,
                        Sequence of Errors OPTIONAL
                        )
```

The optional element “Sequence of Errors” shall be available once at least one of the requested settings was in error. It may be available in order to acknowledge settings explicitly.

Table 2 — MI-SET.confirm parameter description

Name	Description
Link-ID	Unique identifier of a VCI/CI.
CommandRef	Unique cyclic reference number of command.
-	Number of subsequent error messages.
Errors.I-ParamNo	Parameter reference number for which Result.Code applies.
Errors.errStatus	Return/error code, see Table P.1.

A request to set a defined “read only” I-Parameter shall be acknowledged with ErrStatus 7 “ACCESS VIOLATION”. A request to set an undefined I-Parameter shall be acknowledged with ErrStatus 2 “INVALID PARAMETER NUMBER”. A request to set a defined I-Parameter with an illegal value shall be acknowledged with ErrStatus 3 “INVALID PARAMETER VALUE”. A request to set a defined I-Parameter with an illegal type of value shall be acknowledged with ErrStatus 1 “UNSPECIFIED FAILURE”.

An MI-SET.confirm is generated by the MAE upon reception of a previous MI-SET.request.

On receipt of MI-SET.confirm, the ITS-S management entity shall evaluate Errors, if applicable, and act accordingly. Details are outside the scope of this part of ISO 24102.

6.3 MI-GET

6.3.1 MI-GET.request

The service primitive MI-GET.request requests reporting of I-Parameter values to the ITS-S management.

The parameters of the management service primitive MI-GET.request are as follows:

```
MI-GET.request              (
                            Link-ID,
                            CommandRef
                            Sequence of I-ParamNo
                            )
```

Table 3 — MI-GET.request parameter description

Name	Description
Link-ID	Unique identifier of a VCI/CI.
CommandRef	Unique cyclic reference number of command.
-	Number of subsequent elements I-Param.No.
I-ParamNo	See ISO 21218.

This primitive is generated by the ITS-S management entity when I-Parameter values shall be retrieved.

On receipt of MI-GET.request, I-Parameters shall be reported to the ITS-S management entity.

6.3.2 MI-GET.confirm

The management service primitive MI-GET.confirm reports I-Parameter values to the ITS-S management entity.

The parameters of MI-GET.confirm are as follows:

```
MI-GET.confirm         (
                       Link-ID,
                       CommandRef,
                       Sequence of I-Param
                       )
```

Table 4 — MI-GET.confirm parameter description

Name	Description
Link-ID	Unique identifier of a VCI/CI.
CommandRef	Unique cyclic reference number of command.
-	Number of subsequent elements I-Param.
I-Param.No	See ISO 21218.
I-Param.Value	See ISO 21218.

A request to get the value of a defined “write only” I-Parameter shall be acknowledged with ErrStatus 7 “ACCESS VIOLATION”. A request to get the value of an undefined I-Parameter shall be acknowledged with ErrStatus 2 “INVALID PARAMETER NUMBER”.

The management service primitive MI-GET.confirm shall be generated by the MAE upon reception of a previous MI-GET.request.

The ITS-S management entity shall evaluate the parameter values and act accordingly. If any errors occurred in the processing of the related MI-GET.request, the MI-GET.confirm shall contain at least the entry Param.No = 255, where I-Param.Value shall explain the details of the error (see ISO 21218).

Further details are outside the scope of this part of ISO 24102.

6.4 MI-COMMAND

6.4.1 MI-COMMANDs

Annex F provides an overview and coding details on MI-COMMANDs that may be sent by the ITS-S management entity to the ITS-S access layer.

MI-COMMANDs shall be enabled by means of the service primitives MI-COMMAND.request and MI-COMMAND.confirm.

6.4.2 MI-COMMAND.request

The service primitive MI-COMMAND.request allows the ITS-S management to trigger an action at a CI.

The parameters of MI-COMMAND.request are as follows:

```
MI-COMMAND.request          (
                            Link-ID,
                            CommandRef,
                            MI-Command
                            )
```

Table 5 — MI-COMMAND.request parameter description

Name	Description
Link-ID	Unique identifier of a VCI/CI.
CommandRef	Unique cyclic reference number of command.
MI-Command.No	See Annex F.
MI-Command.Value	See Annex F.

An MI-COMMAND.request shall be generated by the ITS-S management when the CI shall perform an action.

On receipt of MI-COMMAND.request by the MAE, the requested action shall be performed.

6.4.3 MI-COMMAND.confirm

The service primitive MI-COMMAND.confirm reports the result of a previous MI-COMMAND.request.

The parameters of MI-COMMAND.confirm are as follows:

```
MI-COMMAND.confirm          (
                            Link-ID,
                            CommandRef
                            ErrStatus
                            )
```

Table 6 — MI-COMMAND.confirm parameter description

Name	Description
Link-ID	Unique identifier of a VCI/CI.
CommandRef	Unique cyclic reference number of command.
ErrStatus	Return/error code, see Table P.1.

An undefined MI-COMMAND shall be acknowledged with ErrStatus 5 “INVALID COMMAND/REQUEST NUMBER”.

An MI-COMMAND.confirm shall be generated by the MAE upon performance of a previous MI-COMMAND.request.

On receipt of MI-COMMAND.confirm, the ITS-S management entity shall evaluate ErrStatus and act accordingly. Details are outside the scope of this part of ISO 24102.

6.5 MI-REQUEST

6.5.1 MI-REQUESTs

Annex G provides an overview and coding details on MI-REQUESTs.

MI-REQUESTs shall be enabled by means of the service primitives MI-REQUEST.request and MI-REQUEST.confirm.

6.5.2 MI-REQUEST.request

The service primitive MI-REQUEST.request allows the MAE to trigger an action at the ITS-S management entity.

The parameters of MI-REQUEST.request are as follows:

```
MI-REQUEST.request          (
                            Link-ID,
                            CommandRef,
                            MI-Request
                            )
```

Table 7 — MI-REQUEST.request parameter description

Name	Description
Link-ID	Unique identifier of a VCI/CI.
CommandRef	Unique cyclic reference number of command.
MI-Request.No	See Annex G.
MI-Request.Value	See Annex G.

An MI-REQUEST.request shall be generated by a MAE when the ITS-S management entity shall perform an action.

On receipt of MI-REQUEST.request the required action is performed.

6.5.3 MI-REQUEST.confirm

The service primitive MI-REQUEST.confirm reports the result of a previous MI-REQUEST.request.

The parameters of MI-REQUEST.confirm are as follows:

```
MI-REQUEST.confirm                  (
                                    Link-ID,
                                    CommandRef,
                                    ErrStatus
                                    )
```

Table 8 — MI-REQUEST.confirm parameter description

Name	Description
Link-ID	Unique identifier of a VCI/CI.
CommandRef	Unique cyclic reference number of command.
ErrStatus	Return/error code, see Table P.1.

An undefined REQUEST shall be acknowledged with ErrStatus 5 “INVALID COMMAND/REQUEST NUMBER”.

An MI-REQUEST.confirm shall be generated by the ITS-S management e upon performance of a previous MI-REQUEST.request.

On receipt of MI-REQUEST.confirm, the MAE shall evaluate ErrStatus and act accordingly. Details are outside the scope of this part of ISO 24102.

6.6 MI-NOTIFY

6.6.1 Pseudo-service

The MI-NOTIFY service is a pseudo-service.

The MI-COMMAND “Monitor” shall be used to start and stop monitoring of a specific I-Parameter.

Notification of an I-Parameter value shall be done with MI-REQUEST “Event”.

6.6.2 Retrieve monitor status

The actual status of monitoring of I-Parameters can be retrieved by means of the MI-GET.request service primitive with I-Parameter “Notify”.

The related MI-GET.response shall show all I-Parameter numbers that are currently activated for monitoring.

6.6.3 MAC management frame

MI-COMMAND "UnitDataCmd" is used by the ITS-S management to request transmission of a management data packet. The MAE shall treat this packet the same way as packets received from the IN-SAP, except for indication of a management packet in the MAC header of the transmitted frame.

MI-REQUEST "UnitDataReq" is used by the MAE to notify reception of a management data packet to the ITS-S management entity.

7 MN-SAP

7.1 General

Basically, there shall be the following different types of services that facilitate

a) sending a command from the ITS-S management entity to the ITS-S networking and transport layer, and

b) receiving a request (command) from the ITS-S networking and transport layer to the ITS-S management entity.

Issuing of commands to the ITS-S networking and transport layer shall be built on the service MN-COMMAND.

Receiving requests from the ITS-S networking and transport layer shall be built on the service MN-REQUEST.

Details of the service primitives shall be as specified in Annex A.

Management communications with the networking and transport layer in other ITS-SCUs shall be with Inter-ITS-SCU communication PDUs "MN-rcmd" and "MN-rreq" as specified in ISO 24102-4.

7.2 MN-COMMAND

7.2.1 MN-COMMANDs

Annex D provides an overview and coding details on MN-COMMANDs that may be sent by the ITS-S management entity to the ITS-S networking and transport layer.

MN-COMMANDs shall be enabled by means of the service primitives MN-COMMAND.request and MN-COMMAND.confirm.

7.2.2 MN-COMMAND.request

The service primitive MN-COMMAND.request allows the ITS-S management entity to trigger an action at the ITS-S networking and transport layer.

The parameters of the management service primitive MN-COMMAND.request are as follows:

```
MN-COMMAND.request          (
                            CommandRef,
                            MN-Command
                            )
```

Table 9 — MN-COMMAND.request parameter description

Name	Description
CommandRef	Unique cyclic reference number of command.
MN-Command.No	Reference number of command.
MN-Command.Value	Value of command.

An MN-COMMAND.request shall be generated by the ITS-S management entity when the ITS-S networking and transport layer shall perform an action.

On receipt of MN-COMMAND.request, the requested action shall be performed.

7.2.3 MN-COMMAND.confirm

The service primitive MN-COMMAND.confirm reports the result of a previous MN-COMMAND.request.

The parameters of the management service primitive MN-COMMAND.confirm are as follows:

```
MN-COMMAND.confirm          (
                            CommandRef,
                            MN-CmdConfirm,
                            ErrStatus
                            )
```

Table 10 — MN-COMMAND.confirm parameter description

Name	Description
CommandRef	Unique cyclic reference number of command. Same value as in related MN-COMMAND.request.
MN-CmdConfirm.No	Reference number of command. Same value as MN-Command.No in related MN-COMMAND.request.
MN-CmdConfirm.Value	Carries optional confirm data.
ErrStatus	Error/return code, see Table P.1.

An MN-COMMAND.confirm shall be generated by the ITS-S networking and transport layer upon performance of a previous MN-COMMAND.request in case ErrStatus indicates error or MN-CmdConfirm is present. It may be generated in case ErrStatus indicates success or MN-CmdConfirm is not present.

On receipt of MN-COMMAND.confirm, ErrStatus and MN-CmdConfirm shall be evaluated and a possible action shall be performed properly.

7.3 MN-REQUEST

7.3.1 MN-REQUESTs

Annex E provides an overview and coding details on MN-REQUESTs.

MN-REQUESTs shall be enabled by means of the service primitives MN-REQUEST.request and MN-REQUEST.confirm.

7.3.2 MN-REQUEST.request

The service primitive MN-REQUEST.request allows the ITS-S networking and transport layer to trigger an action at the ITS-S management entity.

The parameters of MN-REQUEST.request are as follows:

```
MN-REQUEST.request          (
                            CommandRef,
                            MN-Request
                            )
```

Table 11 — MN-REQUEST.request parameter description

Name	Description
CommandRef	Unique cyclic reference number of command.
MN-Request.No	Reference number of the request.
MN-Request.Value	Value of the request.

An MN-REQUEST.request shall be generated by the ITS-S networking and transport layer when the ITS-S management entity shall perform an action.

On receipt of MN-REQUEST.request, the required action shall be performed.

7.3.3 MN-REQUEST.confirm

The service primitive MN-REQUEST.confirm reports the result of a previous MN-REQUEST.request.

The parameters of MN-REQUEST.confirm are as follows:

```
MN-REQUEST.confirm          (
                            CommandRef,
                            MN-ReqConfirm,
                            ErrStatus
                            )
```

Table 12 — MN-REQUEST.confirm parameter description

Name	Description
CommandRef	Unique cyclic reference number of command. Same value as in related MN-REQUEST. request.
MN-ReqConfirm.No	Reference number of command. Same value as MN-Request.No in related MN-REQUEST.request.
MN-ReqConfirm.Value	Carries optional confirm data.
ErrStatus	Error/return code, see Table P.1.

The service primitive MN-REQUEST.confirm shall be generated by the ITS station management entity upon performance of a previous MN-REQUEST.request in case ErrStatus indicates error or MN-ReqConfirm is present. It may be generated in case ErrStatus indicates success or MN-ReqConfirm is not present.

On receipt of MN-REQUEST.confirm, ErrStatus and MN-ReqConfirm shall be evaluated and a possible action shall be performed properly. Details are outside the scope of this part of ISO 24102.

8 MF-SAP

8.1 General

Basically, there shall be the following different types of services that facilitate

a) sending a command from the ITS-S management entity to the ITS-S facilities layer, and

b) receiving a request (command) from the ITS-S facilities layer by the ITS-S management entity.

Issuing of commands to the facilities layer shall be built on the service MF-COMMAND.

Receiving of requests from the facilities layer shall be built on the service MF-REQUEST.

Details of the service primitives shall be as specified in Annex A.

Management communications with other ITS-SCUs shall be with Inter-ITS-SCU communication PDUs “MF-rcmd” and “MF-rreq” as specified in ISO 24102-4.

8.2 MF-COMMAND

8.2.1 MF-COMMANDs

Annex B provides an overview and coding details on MF-COMMANDs that may be sent by the ITS-S management entity to the facilities layer.

MF-COMMANDs shall be enabled by means of the service primitives MF-COMMAND.request and MF-COMMAND.confirm.

8.2.2 MF-COMMAND.request

The service primitive MF-COMMAND.request allows the ITS-S management entity to trigger an action at the ITS-S facilities layer.

The parameters of MF-COMMAND.request are as follows:

```
MF-COMMAND.request          (
                            CommandRef,
                            MF-Command
                            )
```

Table 13 — MF-COMMAND.request parameter description

Name	Description
CommandRef	Unique cyclic reference number of command.
MF-Command.No	Reference number of command.
MF-Command.Value	Value of command.

An MF-COMMAND.request shall be generated by the ITS-S management entity when the ITS-S facilities layer shall perform an action.

On receipt of MF-COMMAND.request the requested action shall be performed.

8.2.3 MF-COMMAND.confirm

The service primitive MF-COMMAND.confirm reports the result of a previous MF-COMMAND.request.

The parameters of MF-COMMAND.confirm are as follows:

```
MF-COMMAND.confirm          (
                            CommandRef,
                            MF-CmdConfirm,
                            ErrStatus
                            )
```

Table 14 — MF-COMMAND.confirm parameter description

Name	Description
CommandRef	Unique cyclic reference number of command. Same value as in related MF-COMMAND.request.
MF-CmdConfirm.No	Reference number of command. Same value as MF-Command.No in related MF-COMMAND.request.
MF-CmdConfirm.Value	Carries optional confirm data.
ErrStatus	Error/return code, see Table P.1.

An MF-COMMAND.confirm shall be generated by the ITS-S facilities layer upon performance of a previous MF-COMMAND.request in case ErrStatus indicates error or MF-CmdConfirm is present. It may be generated in case ErrStatus indicates success or MF-CmdConfirm is not present.

On receipt of this primitive, ErrStatus and MF-CmdConfirm shall be evaluated and a possible action shall be performed properly.

8.3 MF-REQUEST

8.3.1 MF-REQUESTs

Annex C provides an overview and coding details on MF-REQUESTs.

MF-REQUESTs shall be enabled by means of the service primitives MF-REQUEST.request and MF-REQUEST.confirm.

8.3.2 MF-REQUEST.request

The service primitive MF-REQUEST.request allows the ITS-S facilities layer to trigger an action at the ITS-S management entity.

The parameters of MF-REQUEST.request are as follows:

```
MF-REQUEST.request          (
                            CommandRef,
                            MN-Request
                            )
```

Table 15 — MF-REQUEST.request parameter description

Name	Description
CommandRef	Unique cyclic reference number of command.
MF-Request.No	Reference number of the request.
MF-Request.Value	Value of the request.

An MF-REQUEST.request shall be generated by the facilities layer when the ITS-S management entity shall perform an action.

On receipt of MF-REQUEST.request, the required action shall be performed.

8.3.3 MF-REQUEST.confirm

The service primitive MF-REQUEST.confirm reports the result of a previous MF-REQUEST.request.

The parameters of MF-REQUEST.confirm are as follows:

```
MF-REQUEST.confirm          (
                            CommandRef,
                            MF-ReqConfirm,
                            ErrStatus
                            )
```

Table 16 — MF-REQUEST.confirm parameter description

Name	Description
CommandRef	Unique cyclic reference number of command. Same value as in related MF-REQUEST.request.
MF-ReqConfirm.No	Reference number of command. Same value as MF-Request.No in related MF-REQUEST.request.
MF-ReqConfirm.Value	Carries optional confirm data.
ErrStatus	Error/return code, see Table P.1.

An MF-REQUEST.confirm shall be generated by the ITS-S management entity upon performance of a previous MF-REQUEST.request in case ErrStatus indicates error or MF-ReqConfirm is present. It may be generated in case ErrStatus indicates success or MF-ReqConfirm is not present.

On receipt of MF-REQUEST.confirm, ErrStatus and MF-ReqConfirm shall be evaluated and a possible action shall be performed properly. Details are outside the scope of this part of ISO 24102.

9 SI-SAP

9.1 General

Basically, there are the following different types of services:

a) a service that allows the ITS-S security entity to send a command to a CI / VCI in the ITS-S access layer; and

b) a service that allows a CI / VCI in the ITS-S access layer to send a command to the ITS-S security entity.

Issuing of commands by a protocol in the ITS-S security entity shall be built on the service SI-COMMAND.

Issuing of commands by a protocol in the ITS-S access layer, i.e. by a CI or VCI, shall be built on the service SI-REQUEST.

Every request and confirm service primitive shall be identified by a CommandRef parameter. A value unique within the available number space shall be assigned to the request service primitive. The related confirm service primitive shall use the same number as provided in the related request service primitive.

Details of the service primitives shall be as specified in Annex A.

9.2 SI-COMMAND

9.2.1 SI-COMMANDs

Annex L provides an overview and coding details on SI-COMMANDs that may be sent by the ITS-S security entity to the ITS-S access layer.

SI-COMMANDs shall be enabled by means of the service primitives SI-COMMAND.request and SI-COMMAND.confirm.

9.2.2 SI-COMMAND.request

The service primitive SI-COMMAND.request allows the ITS-S security entity to trigger an action at a CI / VCI.

The parameters of SI-COMMAND.request are as follows:

```
SI-COMMAND.request          (
                            Link-ID,
                            CommandRef,
                            SI-Command
                            )
```

Table 17 — SI-COMMAND.request parameter description

Name	Description
Link-ID	Unique identifier of a VCI/CI.
CommandRef	Unique cyclic reference number of command.
SI-Command.No	See Annex L.
SI-Command.Value	See Annex L.

An SI-COMMAND.request shall be generated by the ITS-S security entity when the CI shall perform an action.

On receipt of a SI-COMMAND.request by a CI / VCI, the requested action shall be performed.

9.2.3 SI-COMMAND.confirm

The service primitive SI-COMMAND.confirm reports the result of a previous SI-COMMAND.request.

The parameters of SI-COMMAND.confirm are as follows:

```
SI-COMMAND.confirm          (
                            Link-ID,
                            CommandRef
                            ErrStatus
                            )
```

Table 18 — SI-COMMAND.confirm parameter description

Name	Description
Link-ID	Unique identifier of a VCI/CI.
CommandRef	Unique cyclic reference number of command.
ErrStatus	Return/error code, see Table P.1.

An undefined SI-COMMAND shall be acknowledged with ErrStatus 5 “INVALID COMMAND/REQUEST NUMBER”.

An SI-COMMAND.confirm shall be generated by a CI/VCI upon performance of a previous SI-COMMAND.request.

On receipt of SI-COMMAND.confirm, the ITS security entity shall evaluate ErrStatus and act accordingly. Details are outside the scope of this part of ISO 24102.

9.3 SI-REQUEST

9.3.1 S-REQUESTs

Annex M provides an overview and coding details on SI-REQUESTs.

SI-REQUESTs shall be enabled by means of the service primitives SI-REQUEST.request and SI-REQUEST.confirm.

9.3.2 SI-REQUEST.request

The service primitive SI-REQUEST.request allows a CI/VCI to trigger an action at the ITS-S security entity.

The parameters of SI-REQUEST.request are as follows:

```
SI-REQUEST.request          (
                            Link-ID,
                            CommandRef,
                            SI-Request
                            )
```

Table 19 — SI-REQUEST.request parameter description

Name	Description
Link-ID	Unique identifier of a VCI/CI.
CommandRef	Unique cyclic reference number of command.
SI-Request.No	See Annex M.
SI-Request.Value	See Annex M.

An SI-REQUEST.request shall be generated by a CI/VCI when the ITS-S security entity shall perform an action.

On receipt of a SI-REQUEST.request, the required action shall be performed.

9.3.3 SI-REQUEST.confirm

The service primitive SI-REQUEST.confirm reports the result of a previous SI-REQUEST.request.

The parameters of SI-REQUEST.confirm are as follows:

```
SI-REQUEST.confirm          (
                            Link-ID,
                            CommandRef,
                            ErrStatus
                            )
```

Table 20 — SI-REQUEST.confirm parameter description

Name	Description
Link-ID	Unique identifier of a VCI/CI.
CommandRef	Unique cyclic reference number of command.
ErrStatus	Return/error code, see Table P.1.

An undefined REQUEST shall be acknowledged with ErrStatus 5 “INVALID COMMAND/REQUEST NUMBER”.

An MI-REQUEST.confirm shall be generated by the ITS-S security entity upon performance of a previous SI-REQUEST.request.

On receipt of MI-REQUEST, the CI/VCI shall evaluate ErrStatus and act accordingly. Details are outside the scope of this part of ISO 24102.

10 SN-SAP

10.1 General

Basically, there shall be the following different types of services that facilitate

a) sending a command from the ITS-S security entity to the ITS-S networking and transport layer, and

b) receiving a request (command) from the ITS-S networking and transport layer by the ITS-S security entity.

Issuing of commands to the ITS-S networking and transport layer shall be built on the service SN-COMMAND.

Receiving requests from the ITS-S networking and transport layer shall be built on the service SN-REQUEST.

Details of the service primitives shall be as specified in Annex A.

10.2 SN-COMMAND

10.2.1 SN-COMMANDs

Annex J provides an overview and coding details on SN-COMMANDs that may be sent by the ITS-S security entity to the ITS-S networking and transport layer.

SN-COMMANDs shall be enabled by means of the service primitives SN-COMMAND.request and SN-COMMAND.confirm.

10.2.2 SN-COMMAND.request

The service primitive SN-COMMAND.request allows the ITS-S security entity to trigger an action at the ITS-S networking and transport layer.

The parameters of SN-COMMAND.request are as follows:

```
SN-COMMAND.request          (
                            CommandRef,
                            SN-Command
                            )
```

Table 21 — SN-COMMAND.request parameter description

Name	Description
CommandRef	Unique cyclic reference number of command.
SN-Command.No	See Annex J.
SN-Command.Value	See Annex J.

An SN-COMMAND.request shall be generated by the ITS-S security entity when the ITS-S networking and transport layer shall perform an action.

On receipt of an SN-COMMAND.request, the requested action shall be performed.

10.2.3 SN-COMMAND.confirm

The service primitive SN-COMMAND.confirm reports the result of a previous SN-COMMAND.request.

The parameters of SN-COMMAND.confirm are as follows:

```
SN-COMMAND.confirm          (
                            CommandRef,
                            SN-CmdConfirm,
                            ErrStatus
                            )
```

Table 22 — SN-COMMAND.confirm parameter description

Name	Description
CommandRef	Unique cyclic reference number of command. Same value as in related SN-COMMAND.request.
SN-CmdConfirm.No	Reference number of command. Same value as SN-Command.No in related SN-COMMAND.request.
SN-CmdConfirm.Value	Carries optional confirm data.
ErrStatus	Error/return code, see Table P.1.

An SN-COMMAND.confirm shall be generated by the ITS-S networking and transport layer upon performance of a previous SN-COMMAND.request in case ErrStatus indicates error or SN-CmdConfirm is present. It may be generated in case ErrStatus indicates success or SN-CmdConfirm is not present.

On receipt of SN-COMMAND.confirm, ErrStatus and SN-CmdConfirm shall be evaluated and a possible action shall be performed properly.

10.3 SN-REQUEST

10.3.1 SN-REQUESTs

Annex K provides an overview and coding details on SN-REQUESTs.

SN-REQUESTs shall be enabled by means of the service primitives SN-REQUEST.request and SN-REQUEST.confirm.

10.3.2 SN-REQUEST.request

The security service primitive SN-REQUEST.request allows the ITS-S networking and transport layer to trigger an action at the ITS-S security entity.

The parameters of SN-REQUEST.request are as follows:

```
SN-REQUEST.request          (
                            CommandRef,
                            SN-Request
                            )
```

Table 23 — SN-REQUEST.request parameter description

Name	Description
CommandRef	Unique cyclic reference number of command.
SN-Request.No	See Annex K.
SN-Request.Value	See Annex K.

An SN-REQUEST.request shall be generated by the ITS-S networking and transport layer when the ITS-S security entity shall perform an action.

On receipt of SN-REQUEST.request, the required action shall be performed.

10.3.3 SN-REQUEST.confirm

The service primitive SN-REQUEST.confirm reports the result of a previous SN-REQUEST.request.

The parameters of SN-REQUEST.confirm are as follows:

```
SN-REQUEST.confirm          (
                            CommandRef,
                            SN-ReqConfirm,
                            ErrStatus
                            )
```

Table 24 — SN-REQUEST.confirm parameter description

Name	Description
CommandRef	Unique cyclic reference number of command. Same value as in related SN-REQUEST.request.
SN-ReqConfirm.No	Reference number of command. Same value as SN-Request.No in related SN-REQUEST.request.
SN-ReqConfirm.Value	Carries optional confirm data.
ErrStatus	Error/return code, see Table P.1.

An SN-REQUEST.confirm shall be generated by the ITS-S security entity upon performance of a previous SN-REQUEST.request in case ErrStatus indicates error or SN-ReqConfirm is present. It may be generated in case ErrStatus indicates success or SN-ReqConfirm is not present.

On receipt of SN-REQUEST.confirm, ErrStatus and SN-ReqConfirm shall be evaluated and a possible action shall be performed properly. Details are outside the scope of this part of ISO 24102.

11 SF-SAP

11.1 General

Basically, there shall be the following different types of services that facilitate

a) sending a command from the ITS-S security entity to the ITS-S facilities layer, and

b) receiving a request (command) from the ITS-S facilities layer by the ITS-S security entity.

Issuing of commands by the ITS-S security entity to the ITS-S facilities layer shall be built on the service SF-COMMAND.

Receiving of requests from the ITS-S facilities layer by the ITS-S security entity shall be built on the service SF-REQUEST.

Details of the service primitives shall be as specified in Annex A.

11.2 SF-COMMAND

11.2.1 SF-COMMANDs

Annex H provides an overview and coding details on SF-COMMANDs that may be sent by the ITS-S security entity to the ITS-S facilities layer.

SF-COMMANDs shall be enabled by means of the service primitives SF-COMMAND.request and SF-COMMAND.confirm.

11.2.2 SF-COMMAND.request

The service primitive SF-COMMAND.request allows the ITS-S security entity to trigger an action at the ITS-S facilities layer.

The parameters of SF-COMMAND.request are as follows:

```
SF-COMMAND.request          (
                            CommandRef,
                            SF-Command
                            )
```

Table 25 — SF-COMMAND.request parameter description

Name	Description
CommandRef	Unique cyclic reference number of command.
SF-Command.No	See Annex H.
SF-Command.Value	See Annex H.

An SF-COMMAND.request shall be generated by the ITS-S security entity when the ITS-S facilities layer shall perform an action.

On receipt of SF-COMMAND.request, the requested action shall be performed.

11.2.3 SF-COMMAND.confirm

The service primitive SF-COMMAND.confirm reports the result of a previous SF-COMMAND.request.

The parameters of SF-COMMAND.confirm are as follows:

```
SF-COMMAND.confirm          (
                            CommandRef,
                            SF-CmdConfirm,
                            ErrStatus
                            )
```

Table 26 — SF-COMMAND.confirm parameter description

Name	Description
CommandRef	Unique cyclic reference number of command. Same value as in related SF-COMMAND.request.
SF-CmdConfirm.No	Reference number of command. Same value as SF-Command.No in related SF-COMMAND.request.
SF-CmdConfirm.Value	Carries optional confirm data.
ErrStatus	Error/return code, see Table P.1.

An SF-COMMAND.confirm shall be generated by the ITS-S facilities layer upon performance of a previous SF-COMMAND.request in case ErrStatus indicates error or SF-CmdConfirm is present. It may be generated in case ErrStatus indicates success or SMF-CmdConfirm is not present.

On receipt of SF-COMMAND.confirm, ErrStatus and SF-CmdConfirm shall be evaluated and a possible action shall be performed properly.

11.3 SF-REQUEST

11.3.1 SF-REQUESTs

Annex I provides an overview and coding details on SF-REQUESTs.

SF-REQUESTs shall be enabled by means of the service primitives SF-REQUEST.request and SF-REQUEST.confirm.

11.3.2 SF-REQUEST.request

The service primitive MF-REQUEST.request allows the ITS-S facilities layer to trigger an action at the ITS-S security entity.

The parameters of MF-REQUEST.request are as follows:

```
SF-REQUEST.request          (
                           CommandRef,
                           SN-Request
                           )
```

Table 27 — SF-REQUEST.request parameter description

Name	Description
CommandRef	Unique cyclic reference number of command.
SF-Request.No	See Annex I.
SF-Request.Value	See Annex I.

An SF-REQUEST.request shall be generated by the facilities layer when the ITS station management entity shall perform an action.

On receipt of SF-REQUEST.request, the required action shall be performed.

11.3.3 SF-REQUEST.confirm

The service primitive SF-REQUEST.confirm reports the result of a previous SF-REQUEST.request.

The parameters of SF-REQUEST.confirm are as follows:

```
SF-REQUEST.confirm          (
                           CommandRef,
                           SF-ReqConfirm,
                           ErrStatus
                           )
```

Table 28 — SF-REQUEST.confirm parameter description

Name	Description
CommandRef	Unique cyclic reference number of command. Same value as in related SF-REQUEST.request.
SF-ReqConfirm.No	Reference number of command. Same value as SF-Request.No in related SF-REQUEST.request.

Table 28 *(continued)*

Name	Description
SF-ReqConfirm.Value	Carries optional confirm data.
ErrStatus	Error/return code, see Table P.1.

An SF-REQUEST.confirm shall be generated by the ITS-S security entity upon performance of a previous SF-REQUEST.request in case ErrStatus indicates error or SF-ReqConfirm is present. It may be generated in case ErrStatus indicates success or MF-ReqConfirm is not present.

On receipt of SF-REQUEST.confirm, ErrStatus and SF-ReqConfirm shall be evaluated and a possible action shall be performed properly. Details are outside the scope of this part of ISO 24102.

12 MS-SAP

12.1 General

Basically, there shall be the following different types of services that facilitate

a) sending a command from the ITS-S management entity to the ITS-S security entity, and

b) receiving a request (command) from the ITS-S security entity by the ITS-S management entity.

Issuing of commands by the ITS-S management entity to the ITS-S security entity shall be built on the service MS-COMMAND.

Receiving of requests from the ITS-S security entity by the ITS-S management entity shall be built on the service MS-REQUEST.

Details of the service primitives shall be as specified in Annex A.

12.2 MS-COMMAND

12.2.1 MS-COMMANDs

Annex N provides an overview and coding details on MS-COMMANDs that may be sent by the ITS-S management entity to the ITS-S security entity.

MS-COMMANDs shall be enabled by means of the service primitives MS-COMMAND.request and MS-COMMAND.confirm.

12.2.2 MS-COMMAND.request

The management service primitive MS-COMMAND.request allows the ITS-S management entity to trigger an action at the ITS-S security entity.

The parameters of the management service primitive MS-COMMAND.request are as follows:

```
MS-COMMAND.request          (
                            CommandRef,
                            MS-Command
                            )
```

Table 29 — MS-COMMAND.request parameter description

Name	Description
CommandRef	Unique cyclic reference number of command.
MS-Command.No	See Annex N.
MS-Command.Value	See Annex N.

An MS-COMMAND.request shall be generated by the ITS-S management entity when the ITS-S security entity shall perform an action.

On receipt of MS-COMMAND.request, the requested action shall be performed.

12.2.3 MS-COMMAND.confirm

The service primitive MS-COMMAND.confirm reports the result of a previous MS-COMMAND.request.

The parameters of MS-COMMAND.confirm are as follows:

```
MS-COMMAND.confirm          (
                            CommandRef,
                            MS-CmdConfirm,
                            ErrStatus
                            )
```

Table 30 — MS-COMMAND.confirm parameter description

Name	Description
CommandRef	Unique cyclic reference number of command. Same value as in related MS-COMMAND.request.
MS-CmdConfirm.No	Reference number of command. Same value as MS-Command.No in related MS-COMMAND.request.
MS-CmdConfirm.Value	Carries optional confirm data.
ErrStatus	Error/return code, see Table P.1.

An MS-COMMAND.confirm shall be generated by the ITS-S security entity upon performance of a previous MS-COMMAND.request in case ErrStatus indicates error or MS-CmdConfirm is present. It may be generated in case ErrStatus indicates success or MS-CmdConfirm is not present.

On receipt of MS-COMMAND.confirm, ErrStatus and MS-CmdConfirm shall be evaluated and a possible action shall be performed properly.

12.3 MS-REQUEST

12.3.1 MS-REQUESTs

Annex O provides an overview and coding details on MS-REQUESTs.

MS-REQUESTs shall be enabled by means of the service primitives MS-REQUEST.request and MS-REQUEST.confirm.

12.3.2 MS-REQUEST.request

The management service primitive MS-REQUEST.request allows the ITS-S security entity to trigger an action at the ITS-S management entity.

The parameters of MS-REQUEST.request are as follows:

```
MS-REQUEST.request          (
                            CommandRef,
                            MS-Request
                            )
```

Table 31 — MS-REQUEST.request parameter description

Name	Description
CommandRef	Unique cyclic reference number of command.
MS-Request.No	See Annex O.
MS-Request.Value	See Annex O.

An MS-REQUEST.request shall be generated by the ITS-S security entity when the ITS station management entity shall perform an action.

On receipt of MS-REQUEST.request, the required action shall be performed.

12.3.3 MS-REQUEST.confirm

The service primitive MS-REQUEST.confirm reports the result of a previous MS-REQUEST.request.

The parameters of MS-REQUEST.confirm are as follows:

```
MS-REQUEST.confirm          (
                            CommandRef,
                            MS-ReqConfirm,
                            ErrStatus
                            )
```

Table 32 — MS-REQUEST.confirm parameter description

Name	Description
CommandRef	Unique cyclic reference number of command. Same value as in related MS-REQUEST.request.
MS-ReqConfirm.No	Reference number of command. Same value as MS-Request.No in related MS-REQUEST.request.
MS-ReqConfirm.Value	Carries optional confirm data.
ErrStatus	Error/return code, see Table P.1.

An MS-REQUEST.confirm shall be generated by the ITS-S management entity upon performance of a previous MS-REQUEST.request in case ErrStatus indicates error or MS-ReqConfirm is present. It may be generated in case ErrStatus indicates success or MS-ReqConfirm is not present.

On receipt of MS-REQUEST.confirm, ErrStatus and MS-ReqConfirm shall be evaluated and a possible action shall be performed properly. Details are outside the scope of this part of ISO 24102.

13 Conformance

Service access points may be not observable and not testable. Thus, no conformance tests are specified.

NOTE Service access points may become observable and thus testable as PDUs for Inter-ITS-SCU communications specified in ISO 24102-4.

Annex A
(normative)

ASN.1 modules

A.1 Overview

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

— CALMmsap { ISO (1) standard (0) calm-management (24102) msap (3) version1 (1)}

A.2 Module CALMmsap

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 and enabling cheap implementations, “fill” bits were defined. All fill bits shall be set to the value ‘0’b. In case fill bits precede a CHOICE type, the CHOICE tag shall be evaluated only in case all fill bits show the value ‘0’b in order to identify a conflict with newer versions of the specification providing more choices.

```
CALMmsap { iso (1) standard (0) calm-management (24102) msap (3) version1 (1)}

DEFINITIONS AUTOMATIC TAGS::=BEGIN

IMPORTS

CIclass, CIstatus, Errors, INsapPrimitivesDown, INsapPrimitivesUp, I-Param, I-ParamNo,
KineVectOut, Link-ID, MACaddress, MedID, MedType, UserPriority FROM CALMllsap {iso(1)
standard(0) calm-ll-sap(21218) version1(1)}

ITS-scuId, ITS-SSI FROM CALMmanagement { iso (1) standard (0) calm-management (24102)
local(1) version1 (1)}

IICrequestTX, IICresponseTX, IICrequestRX, IICresponseRX FROM CALMiitsscu { iso (1)
standard (0) calm-management (24102) iitsscu (4) version1 (1)}

GCctx, GCdeleteCmd, GCregServer, GCregServerConf, GCsam, GCsamctx, GCsamctxConf,
GCctxTxCmd, GCperiodCmd, CTXrxNot, SAMrxNot, GCupdateServer, GCupdateServerConf,
GCderegServer, GCderegServerConf, GCregClient, GCregClientConf, GCupdateClient,
GCupdateClientConf, GCderegClient, GCderegClientConf FROM CALMfsap { iso (1) standard (0)
calm-management (24102) fsap (5) version1 (1)}

SetIPv6, SetConfIPv6, UpdateIPv6, DeleteIPv6, SetNotIPv6, UpdateNotIPv6, DeleteNotIPv6
FROM CALMipv6{iso(1) standard(0) calm-ipv6(21210) version0(0)}

HostServiceInfo, NFsapPrimitivesDown, NFsapPrimitivesUp, SetFNTP, SetConfFNTP, UpdateFNTP,
DeleteFNTP, SetNotFNTP, UpdateNotFNTP, DeleteNotFNTP FROM CALMfntp { iso (1) standard (0)
calm-nonip(29281) fntp(1) version1 (1)}

ITS-S-Appl-Reg FROM CITSapplReq {iso(1) standard(0) cits-applReq (17423) version1 (1)}

;
-- End of IMPORTS
-- Types

-- MF-SAP Service primitives --

MF-Command-request::=SEQUENCE{
     commandRef      CommandRef,
     command-param     MF-Command
     }

MF-Command::=SEQUENCE{
     fill           BIT STRING (SIZE(4)),
     mfCmd          CHOICE{
          gCsamctx          GCsamctx,
          gCsam             GCsam,
          gCctx             GCctx,
          lDMnotify         LDMnotify,
          legacyCI          LegacyCIreq,
          stateCInotify     StateCInotify,
          gCperiodCmd       GCperiodCmd,
          gCctxTxCmd        GCctxTxCmd,
          gCdeleteCmd       GCdeleteCmd,
          iICrequestTX      IICrequestTX,
          iICresponseTX     IICresponseTX
          }
     }

LDMnotify::=SEQUENCE (SIZE(0..255)) OF RadarView

RadarView::=SEQUENCE{
     peerITS-SSI   ITS-SSI
     }

LegacyCIreq::=SEQUENCE{
     cIclass        CIclass,
     legacyOption   INTEGER(0..255),
     linkId         Link-ID
     }

StateCInotify::=SEQUENCE{
     linkId         Link-ID,
     cIstatus       CIstatus
}

MF-Command-confirm::=SEQUENCE{
     commandRef     CommandRef,
     cmdConfirm     MF-CmdConfirm,
     errStatus      ErrStatus
     }

MF-CmdConfirm::=SEQUENCE{
     fill          BIT STRING (SIZE(4)),
     mfCmdConf     CHOICE{
          gCsamctx          GCsamctxConf,
          gCsam             NULL,
          gCctx             NULL,
          lDMnotify         NULL,
          legacyCI          HostServiceInfo,
          stateCInotify     NULL,
          gCperiodCmdConf   NULL,
          gCctxTxCmdConf    NULL,
          gCdeleteCmdConf   NULL,
          iICrequestTXConf  NULL,
          iICresponseTXConf NULL
          }
     }

MF-Request-request::=SEQUENCE{
     commandRef        CommandRef,
     request-param     MF-Request
     }

MF-Request::=SEQUENCE{
     fill           BIT STRING (SIZE(4)),
     mfReq          CHOICE{
         iTS-S-Appl-Reg     ITS-S-Appl-Reg,
         gCregServer        GCregServer,
         gCupdateServer     GCupdateServer,
         gCderegServer      GCderegServer,
         gCregClient        GCregClient,
         gCupdateClient     GCupdateClient,
         gCderegClient      GCderegClient,
         lDMregister        LDMregister,
         sAMrxNot           SAMrxNot,
         cTXrxNot           CTXrxNot,
         iICrequestRX       IICrequestRX,
         iICresponseRX      IICresponseRX
         }
     }

LDMregister::=SEQUENCE{
     iTS-scuId     ITS-scuId,
     reference     OCTET STRING (SIZE(0..65535))
     }

MF-Request-confirm::=SEQUENCE{
     commandRef   CommandRef,
     reqConfirm   MF-ReqConfirm,
     errStatus    ErrStatus
     }

MF-ReqConfirm::=SEQUENCE{
     fill         BIT STRING (SIZE(4)),
     mfReqConf    CHOICE{
          iTS-S-Appl-Reg    NULL,
          gCregServer       GCregServerConf,
          gCupdateServer    GCupdateServerConf,
          gCderegServer     GCderegServerConf,
          gCregClient       GCregClientConf,
          gCupdateClient    GCupdateClientConf,
          gCderegClient     GCderegClientConf,
          lDMregister       ITS-scuId,
          sAMrxNot          NULL,
          cTXrxNot          NULL,
          iICrequestRXConf  NULL,
          iICresponseRXConf NULL
          }
     }

-- MN-SAP Service primitives --

MN-Command-request::=SEQUENCE{
     commandRef    CommandRef,
     command-param MN-Command
     }

MN-Command::=SEQUENCE{
     fill         BIT STRING (SIZE(5)),
     mnCmd        CHOICE{
          simNFcmd            SimNFcmd,
          fWTset              FWTset,
          fWTupdate           FWTupdate,
          fWTdelete           FWTdelete
          }
     }

SimNFcmd::=SEQUENCE{
     fill         BIT STRING (SIZE(7)),
     simNFcmd     CHOICE{
          fntp          NFsapPrimitivesDown
          }
    }

FWTset::=SEQUENCE{
     fill         BIT STRING (SIZE(7)),
     set          CHOICE{
          ipv6         SetIPv6,
          fntp         SetFNTP
--        geo          SetGeo,
--        oem          SetOEM,
          }
     }

FWTupdate::=SEQUENCE{
     fill          BIT STRING (SIZE(7)),
     update        CHOICE{
          ipv6          UpdateIPv6,
          fntp          UpdateFNTP
--        geo           UpdateGeo,
--        oem           UpdateOEM,
          }
     }

FWTdelete::=SEQUENCE{
     fill          BIT STRING (SIZE(7)),
     delete        CHOICE{
          ipv6          DeleteIPv6,
          fntp          DeleteFNTP
--        geo           DeleteGeo,
--        oem           DeleteOEM,
          }
    }

MN-Command-confirm::=SEQUENCE{
     commandRef   CommandRef,
     cmdConfirm   MN-CmdConfirm,
     errStatus    ErrStatus
     }

MN-CmdConfirm::=SEQUENCE{
     fill         BIT STRING (SIZE(5)),
     mfCmdConf     CHOICE{
          simNFcmd       NULL,
          fWTsetConf     FWTsetConf,
          fWTupdateConf  NULL,
          fWTdeletConf   NULL
          }
     }

FWTsetConf::=SEQUENCE{
     fill         BIT STRING (SIZE(7)),
     setConf      CHOICE{
          ipv6          SetConfIPv6,
          fast          SetConfFNTP
--        geo           SetConfGeo,
--        oem           SetConfOEM
          }
     }

MN-Request-request::=SEQUENCE{
     commandRef      CommandRef,
     request-param   MN-Request
     }

MN-Request::=SEQUENCE{
     fill          BIT STRING (SIZE(5)),
     mnReq             CHOICE{
          simNFreq           SimNFreq,
          fWTsetNot          FWTsetNot,
          fWTupdateNot       FWTupdateNot,
          fWTdeletNot        FWTdeleteNot,
          vCIcreatePeerMAC   VCIcreatePeerMAC,
          its-ssiPeerNot     Its-ssiPeerNot
          }
     }

SimNFreq::=SEQUENCE{
     fill          BIT STRING (SIZE(7)),
     simNFreq          CHOICE{
          fntp               NFsapPrimitivesUp
          }
    }

FWTsetNot::=SEQUENCE{
     fill              BIT STRING (SIZE(7)),
     setNot            CHOICE{
          ipv6              SetNotIPv6,
          fast              SetNotFNTP
--        geo               SetNotGeo,
--        oem               SetNotOEM,
          }
     }

FWTupdateNot::=SEQUENCE{
     fill              BIT STRING (SIZE(7)),
     updateNot         CHOICE{
          ipv6              UpdateNotIPv6,
          fast              UpdateNotFNTP
--        geo               UpdateNotGeo,
--        oem               UpdateNotOEM,
          }
     }

FWTdeleteNot::=SEQUENCE{
     fill              BIT STRING (SIZE(7)),
     deleteNot         CHOICE{
          ipv6              DeleteNotIPv6,
          fast              DeleteNotFNTP
--        geo               DeleteNotGeo,
--        oem               DeleteNotOEM,
          }
     }

VCIcreatePeerMAC::=SEQUENCE{
     reference         INTEGER(0..255),
     linkId            Link-ID,
     peerMac           MACaddress
     }

Its-ssiPeerNot::=SEQUENCE{
     sap               INTEGER(0..255),
     mACaddress        MACaddress,
     linkId            Link-ID,
     its-ssiData       ITS-SSI
     }

StationType::= INTEGER{
     mobile                 (0),
     fixed                  (1),
     infrastructure         (2),
     vehicle                (3),
     roadside               (4),
     central                (5),
     portable               (6),
     unknown                (255)
     }(0..255)

StationID::=OCTET STRING (SIZE(4))

MN-Request-confirm::=SEQUENCE{
     commandRef      CommandRef,
     reqConfirm      MN-ReqConfirm,
     errStatus       ErrStatus
     }

MN-ReqConfirm::=SEQUENCE{
     fill            BIT STRING (SIZE(5)),
     mnReqConf       CHOICE{
          simNFreq                 NULL,
          fWTsetNot                NULL,
          fWTupdateNot             NULL,
          fWTdeletNot        NULL,
          vCIcreatePeerMAC   VCIPeerMAC,
          its-ssiPeerNot     NULL
          }
     }

VCIPeerMAC::=SEQUENCE{
     reference     INTEGER(0..255),
     linkId        Link-ID
}

-- MI-SAP Service primitives --

MI-Command-request::=SEQUENCE{
     linkID        Link-ID,
     commandRef    CommandRef,
     command-param MI-Command
     }

MI-Command::=SEQUENCE{
     fill          BIT STRING (SIZE(4)),
     miCmd         CHOICE{
          simINcmd           SimINcmd,
          regCmd             RegCmd,
          cIstate            CIstateChng,
          wakeup             WakeUp,
          rTScmd             RTScmd,
          rTSackCmd          RTSackCmd,
          cONcmd             CONcmd,
          rIcmd              RIcmd,
          manuCmd            OCTET STRING,
          vciCmd             VciCmd,
          monitor            Monitor,
          unitDataCmd        UnitData
          }
     }

SimINcmd::=SEQUENCE{
     fill          BIT STRING (SIZE(7)),
     simINcmd      CHOICE{
          inDown        INsapPrimitivesDown
          }
     }

RegCmd::=SEQUENCE{
     scuId         ITS-scuId,
     medID         MedID
     }

CIstateChng::=INTEGER{
     deregister    (0),
     activate      (4),
     resume        (8),
     connect       (16),
     disconnect    (32),
     suspend       (64),
     inactivate    (128)
     }(0..255)

WakeUp::=INTEGER(0..255)

RTScmd::=SEQUENCE{
     reqID         ReqID,
     priority      UserPriority,
     seqNo         INTEGER(0..255),
     status        INTEGER {
                    release       (0), -- release prioritization
                    request       (16) -- request prioritization
                   }(0..255)
    }
```

```
ReqID::=SEQUENCE{
     linkID       Link-ID
     }

RTSackCmd::=SEQUENCE{
     priority     UserPriority,
     seqNo        INTEGER(0..255),
     status       INTEGER {
                   ignored    (64), -- request ignored
                   granted    (128) -- request granted
                  }(0..255)
     }

CONcmd::=INTEGER{
     deleteAC     (0),
     connect      (1),
     disconnect   (255)
     }(0..255)

RIcmd::=SEQUENCE{
     linkID        Link-ID,
     ri            RI
     }

RI::=OCTET STRING (SIZE(0..65535))

VciCmd::=SEQUENCE (SIZE(0..65535)) OF SEQUENCE{
     fill          BIT STRING (SIZE(7)),
     linkID        Link-ID,
     alive         BOOLEAN OPTIONAL
     }

Monitor::=SEQUENCE (SIZE(0..255)) OF SEQUENCE{
     paramNo       INTEGER(0..255),     -- valid parameter number
     active        INTEGER{
                    stop (0),
                    start (255)
                   }(0..255)
     }

MI-Command-confirm::=SEQUENCE{
     linkID        Link-ID,
     commandRef    CommandRef,
     errStatus     ErrStatus
     }

MI-Request-request::=SEQUENCE{
     linkID        Link-ID,
     commandRef    CommandRef,
     request-param MI-Request
     }

MI-Request::=SEQUENCE{
     fill          BIT STRING (SIZE(4)),
     miReq         CHOICE{
          simINreq      SimINreq,
          regReq        RegReq,
          prioReg       PrioReg,
          rTSreq        RTSreq,
          rTSackReq     RTSackReq,
          rIreq         RIreq,
          manure        OCTET STRING,
          events        Events21218,
          posUpdateReq  PosUpdateReq,
          unitDataReq   UnitData
          }
     }

SimINreq::=SEQUENCE{
     fill          BIT STRING (SIZE(7)),
     simINreq      CHOICE{
          inUp          INsapPrimitivesUp
          }
     }

RegReq::=SEQUENCE{
     medType       MedType
     }

PrioReg::=SEQUENCE{
     interferers   Interferers,
     timeout       INTEGER(0..255)
     }

Interferers::=SEQUENCE (SIZE(0..255)) OF MedType

RTSreq::=SEQUENCE{
     priority     UserPriority,
     seqNo        INTEGER(0..255),
     status       INTEGER {
                   release    (0), -- release prioritization
                   request    (16) -- request prioritization
                  }(0..255)
     }

RTSackReq::=SEQUENCE{
     reqID        ReqID,
     seqNo        INTEGER(0..255),
     status       INTEGER {
                   ignored    (64), -- request ignored
                   granted    (128) -- request granted
                  }(0..255)
     }

RIreq::=SEQUENCE{
     medType       MedType,     -- medium for retrieval
     riAccess      RIaccess     -- request frame details
     }

RIaccess::=OCTET STRING (SIZE(0..65535))
   Events21218::=SEQUENCE{
        fill           BIT STRING (SIZE(4)),
        events         CHOICE{
             minUserPrio       E21218-0, -- MinimumUserPriority
             txQueueThreshold  E21218-1, -- TX queue at threshold
             txQueueFull       E21218-2, -- TX queue full
             vciCreated        E21218-3, -- VCI created
             vciDeleted        E21218-4, -- VCI deleted
             paramMonitor      E21218-5, -- Automatic notification
             txQueueLow        E21218-6, -- TX queue below low thresh.
             vciReset          E21218-7 -- VCI reset
             }
        }

E21218-0::=SEQUENCE{
        priority       UserPriority,
        linkID         Link-ID
        }

E21218-1::=UserPriority

E21218-2::=UserPriority

E21218-3::=Link-ID

E21218-4::=Link-ID

E21218-5::=I-Param

E21218-6::=UserPriority

E21218-7::=Link-ID
```

```
PosUpdateReq::=INTEGER(0..65535) -- 0: Update disabled
     -- >0: Update interval in ms

MI-Request-confirm::=SEQUENCE{
     linkID       Link-ID,
     commandRef   CommandRef,
     errStatus    ErrStatus
     }

MI-Set-request::=SEQUENCE{
     linkID       Link-ID,
     commandRef   CommandRef,
     set-param    SEQUENCE OF I-Param
     }

MI-Set-confirm::=SEQUENCE{
     linkID       Link-ID,
     commandRef   CommandRef,
     set-param    SEQUENCE OF Errors
     }

MI-Get-request::=SEQUENCE{
     linkID       Link-ID,
     commandRef   CommandRef,
     get-param-no SEQUENCE OF I-ParamNo
     }

MI-Get-confirm::=SEQUENCE{
     linkID       Link-ID,
     commandRef   CommandRef,
     set-param    SEQUENCE OF I-Param
     }

UnitData::=SEQUENCE{
     sourceAddr   Link-ID,
     destAddr     Link-ID,
     data         OCTET STRING (SIZE(0..65535)),
     priority     UserPriority,
     parameter    OCTET STRING (SIZE(0..65535)) -- tbd dependent on medium
     }

-- SF-SAP Service primitives --

SF-Command-request::=SEQUENCE{
     commandRef      CommandRef,
     command-param   SF-Command
     }

SF-Command::=SEQUENCE{
     fill          BIT STRING (SIZE(7)),
     sfCmd         CHOICE{
          dummy         NULL
          }
     }

SF-Command-confirm::=SEQUENCE{
     commandRef   CommandRef,
     cmdConfirm   SF-CmdConfirm,
     errStatus    ErrStatus
     }

SF-CmdConfirm::=SEQUENCE{
     fill         BIT STRING (SIZE(7)),
     sfCmdConf    CHOICE{
          dummy         NULL
          }
     }

SF-Request-request::=SEQUENCE{
     commandRef        CommandRef,
     request-param     SF-Request
     }

SF-Request::=SEQUENCE{
     fill          BIT STRING (SIZE(7)),
     sfReq         CHOICE{
          dummy         NULL
          }
     }

SF-Request-confirm::=SEQUENCE{
     commandRef   CommandRef,
     reqConfirm   SF-ReqConfirm,
     errStatus    ErrStatus
     }

SF-ReqConfirm::=SEQUENCE{
     fill         BIT STRING (SIZE(7)),
     sfReqConf    CHOICE{
          dummy        NULL
          }
     }

-- SN-SAP Service primitives --

SN-Command-request::=SEQUENCE{
     commandRef      CommandRef,
     command-param   SN-Command
     }

SN-Command::=SEQUENCE{
     fill          BIT STRING (SIZE(7)),
     snCmd         CHOICE{
          dummy         NULL
          }
     }

SN-Command-confirm::=SEQUENCE{
     commandRef   CommandRef,
     cmdConfirm   SN-CmdConfirm,
     errStatus    ErrStatus
     }

SN-CmdConfirm::=SEQUENCE{
     fill         BIT STRING (SIZE(7)),
     snCmdConf    CHOICE{
          dummy        NULL
          }
     }

SN-Request-request::=SEQUENCE{
     commandRef        CommandRef,
     request-param     SN-Request
     }

SN-Request::=SEQUENCE{
     fill          BIT STRING (SIZE(7)),
     snReq         CHOICE{
          dummy         NULL
          }
     }

SN-Request-confirm::=SEQUENCE{
     commandRef   CommandRef,
     reqConfirm   SN-ReqConfirm,
     errStatus    ErrStatus
     }

SN-ReqConfirm::=SEQUENCE{
     fill         BIT STRING (SIZE(7)),
     snReqConf    CHOICE{
          dummy         NULL
          }
     }

-- SI-SAP Service primitives --

SI-Command-request::=SEQUENCE{
     commandRef      CommandRef,
     command-param   SI-Command
     }

SI-Command::=SEQUENCE{
     fill          BIT STRING (SIZE(7)),
     siCmd         CHOICE{
          dummy         NULL
          }
     }

SI-Command-confirm::=SEQUENCE{
     commandRef   CommandRef,
     cmdConfirm   SI-CmdConfirm,
     errStatus    ErrStatus
     }

SI-CmdConfirm::=SEQUENCE{
     fill         BIT STRING (SIZE(7)),
     siCmdConf    CHOICE{
          dummy        NULL
          }
     }

SI-Request-request::=SEQUENCE{
     commandRef       CommandRef,
     request-param    SI-Request
     }

SI-Request::=SEQUENCE{
     fill          BIT STRING (SIZE(7)),
     siReq         CHOICE{
          dummy         NULL
          }
     }

SI-Request-confirm::=SEQUENCE{
     commandRef   CommandRef,
     reqConfirm   SI-CmdConfirm,
     errStatus    ErrStatus
     }

SI-ReqConfirm::=SEQUENCE{
     fill         BIT STRING (SIZE(7)),
     siReqConf    CHOICE{
          dummy        NULL
          }
     }

-- MS-SAP Service primitives --

MS-Command-request::=SEQUENCE{
     commandRef      CommandRef,
     command-param   MS-Command
     }

MS-Command::=SEQUENCE{
     fill          BIT STRING (SIZE(7)),
     msCmd         CHOICE{
          dummy         NULL
          }
     }

MS-Command-confirm::=SEQUENCE{
     commandRef   CommandRef,
     cmdConfirm   MS-CmdConfirm,
     errStatus    ErrStatus
     }

MS-CmdConfirm::=SEQUENCE{
     fill         BIT STRING (SIZE(7)),
     msCmdConf    CHOICE{
          dummy        NULL
          }
     }

MS-Request-request::=SEQUENCE{
     commandRef        CommandRef,
     request-param     MS-Request
     }

MS-Request::=SEQUENCE{
     fill          BIT STRING (SIZE(7)),
     msReq         CHOICE{
          dummy         NULL
          }
     }

MS-Request-confirm::=SEQUENCE{
     commandRef   CommandRef,
     reqConfirm   MS-ReqConfirm,
     errStatus    ErrStatus
     }

MS-ReqConfirm::=SEQUENCE{
     fill         BIT STRING (SIZE(7)),
     msReqConf    CHOICE{
          dummy        NULL
          }
     }

-- General types --

ErrStatus::=INTEGER{
     Success        (0),
     unspecFailure  (1), -- unspecified failure
     invalParamNo   (2), -- invalid param no.
     invalParamVal  (3), -- invalid param value
     riViolation    (4), -- RI violation
     cmdReqNo       (5), -- invalid cmd/req no.
     cmdReqVal      (6), -- invalid cmd/req value
     accessViolation (7), -- access violation
     invalidType    (8), -- invalid cmd/req type
     sequenceError  (9), -- cmd/req not applicable in the given context
     nonavailValue  (10), -- value required in cmd/req is not available
     hardwareFailure (255) -- unspecified hardware failure
     } (0..255)

CommandRef::=INTEGER(0..255)

-- Values


/*
   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)

MF-COMMANDs

B.1 Overview

Table B.1 presents a summary of defined MF-COMMANDs. ASN.1 types are presented for .request/.confirm service primitives. A more detailed description is provided in B.2.

Table B.1 — MF-COMMANDs

COMMAND	Description
GCsamctx	Notifies a service ITS-S application, which is registered for groupcasting, reception of a service advertisement frame (SAF), and acknowledgment of the expected service with a service context frame (SCF). ASN.1 type: `GCsamctx` / -
GCsam	Notifies to a service ITS-S application, which is registered for groupcasting, reception of a service advertisement frame (SAF) without acknowledgment of the expected service with a service context frame (SCF). ASN.1 types: `GCsam` / -
GCctx	Notifies to a server ITS-S application, which is registered for groupcasting, that a SCF requesting a specific context of the service was received. ASN.1 type: `GCctx` / -
LDMnotify	Notifies the LDM application about an update of the “Radar View”. ASN.1 type: `LDMnotify` /-
LegacyCI	Notifies the “Legacy CI Port Manager” of the presence of a “Legacy CI”. ASN.1 type: `LegacyCIreq` / `HostServiceInfo`
StateCInotify	Notifies the state of a CI. ASN.1 type: `StateCInotify` /
GCperiodCmd	Send a groupcast request to the groupcast communication manager in order to trigger subsequent periodic groupcast transmissions to be performed by the FAST networking protocol. ASN.1 type: `GCperiodCmd` / -
GCctxTxCmd	Send “Context message” (CTX) to the groupcast communication manager for the purpose of unicast delivery to the selected peer station. ASN.1 type: `GCctxTxCmd` / -
GCdeleteCmd	Notifies deletion of a registration of an ITS-S application to the groupcast communication manager. ASN.1 type: `GCdeleteCmd` / -
IICrequestTX	Send IIC-Request PDU to IICA for transmission via the ITS station-internal network. ASN.1 type: `IICrequestTX` / -
IICresponseTX	Send IIC-Response PDU to IICA for transmission via the ITS station-internal network. ASN.1 type: `IICresponseTX` / -
SimFAcmd	To be added in a future version of this part of ISO 24102.

B.2 Description

B.2.1 Basics

This subclause defines the detailed structure of MF-COMMANDs.

B.2.2 GCsafscf

This MF-COMMAND shall be as specified in ISO 24102-5.

Optionally, the MF-COMMAND.request may be confirmed with an MF-COMMAND.confirm carrying only an ErrStatus.

B.2.3 GCsam

This MF-COMMAND shall be as specified in ISO 24102-5.

This MF-COMMAND.request shall not be confirmed with an MF-COMMAND.confirm carrying only an ErrStatus.

B.2.4 GCsam

This MF-COMMAND shall be as specified in ISO 24102-5.

This MF-COMMAND.request shall not be confirmed with an MF-COMMAND.confirm carrying only an ErrStatus.

B.2.5 LDMnotify

Table B.2 — MF-COMMAND.request for MF-COMMAND LDMnotify

ASN.1 type	Description
MF-Command.**lDMnotify**	Notification of an update of the “Radar View” to the LDM application.
LDMnotify.**radarView**	Update of “Radar View”.
RadarView.**iTS-SSI**	ITS station state information.
ITS-SSI.**stationType**	See ISO 24102-1.
ITS-SSI.**stationID**	See ISO 24102-1.
ITS-SSI.**kineVectOut**	See ISO 21218.

Optionally, the MF-COMMAND.request may be confirmed with an MF-COMMAND.confirm carrying only an ErrStatus.

B.2.6 LegacyCI

Table B.3 — MF-COMMAND.request for MF-Command LegacyCI

ASN.1 type	Description
MF-Command.**legacyCIrq**	Notification of presence of an active “15628 CI”.
LegacyCIreq.**ciClass**	CI class of legacy CI as specified in ISO 21218.
LegacyCIreq.**legacyOption**	Type of “Legacy CI”.
LegacyCIreq.**linkId**	Link-ID of UC-VCI of the “Legacy CI”.

The MF-COMMAND.request shall be confirmed with an MF-COMMAND.confirm as specified in Table B.4.

Table B.4 — MF-COMMAND.confirm for MF-COMMAND LegacyCI

ASN.1 type	Description
MF-CmdConfirm.**hostServiceInfo**	
HostServiceInfo.**serviceNWref**	ServiceNWref of "Legacy CI Port Manager".
ServiceNWref.**nWref**	NWref pointing to "Legacy CI Port Manager".
ServiceNWref.**ITS-scuId**	ITS-SCU-ID of "Legacy CI Port Manager".

B.2.7 StateCInotify

Table B.5 — MF-COMMAND.request for MF-COMMAND StateCInotify

ASN.1 type	Description
MF-Command.**stateCInotify**	Notification of the state of a CI.
StateCInotify.**linkId**	Link-ID of CI specified in ISO 21218.
StateCInotify.**cIstatus**	Actual state of the CI specified in ISO 21218.

The MF-COMMAND.request shall not be acknowledged with an MF-COMMAND.confirm.

B.2.8 GCperiodCmd

This MF-COMMAND shall be as specified in ISO 24102-5.

B.2.9 GCctxTxCmd

This MF-COMMAND shall be as specified in ISO 24102-5.

B.2.10 GCdeleteCmd

This MF-COMMAND shall be as specified in ISO 24102-5.

B.2.11 IICrequestTX

This MF-COMMAND shall be as specified in ISO 24102-4.

B.2.12 IICresponseTX

This MF-COMMAND shall be as specified in ISO 24102-4.

Annex C
(normative)

MF-REQUESTs

C.1 Overview

Table C.1 presents a summary of defined MF-REQUESTs. ASN.1 types are presented for .request/.confirm service primitives. A more detailed description is provided in C.2.

Table C.1 — MF-REQUESTs

REQUEST name	Description
ITS-S-Appl-Reg	Allows an ITS-S application to announce its communication needs to the CI selection manager. ASN.1 types: `ITS-S-Appl-Reg` / -
GCregServer	Allows a server application to register for groupcasting providing serviceData. ASN.1 type: `GCregServer` / `GCregServerConf`
GCupdateServer	Allows a server application to update groupcast information providing serviceData. ASN.1 type: `GCupdateServer` / `GCupdateServerConf`
GCderegServer	Allows a server application to deregister from groupcasting. ASN.1 type: `GCderegServer` / `GCderegServerConf`
GCregClient	Allows a service user application to register for groupcasting. ASN.1 type: `GCregClient` / `GCregClientConf`
GCupdateClient	Allows a service user application to update groupcast information. ASN.1 type: `GCupdateClient` / `GCregClientConf`
GCderegClient	Allows a service user application to deregister from groupcasting. ASN.1 type: `GCderegClient` / `GCderegClientConf`
LDMregister	Allows the LDM application to register at the ITS station management entity in order to receive updates of the “Radar View”. ASN.1 type: `LDMregister` / `ITS-scuId`
SAMrxNot	Notifies reception of SAM. ASN.1 type: `SAMrxNot` / -
CTXrxNot	Notifies reception of CTX. ASN.1 type: `CTXrxNot` / -
IICrequestRX	Notifies reception of an IIC-Request PDU. ASN.1 type:
IICresponseRX	Notifies reception of an IIC-Response PDU. ASN.1 type:
SimFAreq	To be added in a future version of this part of ISO 24102.

C.2 Description

C.2.1 Basics

This subclause defines the detailed structure of MF-REQUESTs.

C.2.2 ITS-S-Appl-Reg

This MF-REQUEST shall be as specified in ISO 24102-1.

This MF-REQUEST shall be processed by the CI selection manager in the ITS-SCU for local usage and shall be forwarded to all other ITS-SCUs in the same ITS station.

In case the MF-REQUEST was not properly processed, MF-REQUEST.confirm shall indicate the error. Otherwise, there shall not be an acknowledgement of the MF-REQUEST.

C.2.3 GCregServer

This MF-REQUEST shall be as specified in ISO 24102-5.

In case of failure to process the MF-REQUEST.request, it shall be confirmed with an MF-REQUEST.confirm carrying only an ErrStatus.

C.2.4 GCupdateServer

This MF-REQUEST shall be as specified in ISO 24102-5.

In case of failure to process the MF-REQUEST.request, it shall be confirmed with an MF-REQUEST.confirm carrying only an ErrStatus.

C.2.5 GCderegServer

This MF-REQUEST shall be as specified in ISO 24102-5.

In case of failure to process the MF-REQUEST.request, it shall be confirmed with an MF-REQUEST.confirm carrying only an ErrStatus.

C.2.6 GCregClient

This MF-REQUEST shall be as specified in ISO 24102-5.

In case of failure to process the MF-REQUEST.request, it shall be confirmed with an MF-REQUEST.confirm carrying only an ErrStatus.

C.2.7 GCupdateClient

This MF-REQUEST shall be as specified in ISO 24102-5.

In case of failure to process the MF-REQUEST.request, it shall be confirmed with an MF-REQUEST.confirm carrying only an ErrStatus.

C.2.8 GCderegClient

This MF-REQUEST shall be as specified in ISO 24102-5.

In case of failure to process the MF-REQUEST.request, it shall be confirmed with an MF-REQUEST.confirm carrying only an ErrStatus.

C.2.9 LDMregister

Table C.2 — MF-REQUEST.request for MF-REQUEST LDMregister

ASN.1 type	Description
MF-Request.**lDMregister**	Allows the LDM application to register at the ITS station management entity in order to receive updates of the “Radar View”.
LDMregister.**ITS-scuId**	ITS-SCU-ID of host where LDM application resides.
LDMregister.**reference**	Defined by LDM application.

The MF-REQUEST.request shall be confirmed with an MF-REQUEST.confirm as specified in Table C.3.

Table C.3 — MF-REQUEST.confirm for MF-REQUEST LDMregister

ASN.1 type	Description
MF-Request.**lDMregister**	ITS-SCU-ID of router where “Radar View” data are produced.

C.2.10 SAMrxNot

This MF-REQUEST shall be as specified in ISO 24102-5.

C.2.11 CTXrxNot

This MF-REQUEST shall be as specified in ISO 24102-5.

Annex D
(normative)

MN-COMMANDs

D.1 Overview

Table D.1 presents a summary of defined MN-COMMANDs. ASN.1 types are presented for .request/.confirm service primitives. A more detailed description is provided in D.2.

Table D.1 — MN-COMMANDs

COMMAND name	Description
SimNFcmd	Command to simulate an access to the networking and transport layer via NF-SAP. Applicable only in test mode. ASN.1: `SimNFcmd` / -
FWTset	Sets an entry in the forwarding table of a networking protocol. ASN.1: `FWTset` / `FWTsetConf`
FWTupdate	Updates an entry in the forwarding table of a networking protocol. ASN.1: `FWTupdate` / `FWTupdateNot`
FWTdelete	Deletes an entry in the forwarding table of a networking protocol. ASN.1: `FWTdelete` / `FWTdeleteNot`
PathMgmt	Command for managing a path. NOTE Details will be specified in the next revision of this part of ISO 24102.
FlowClassificationRule	Attribute an identifier to each packet that is output or forwarded by ITS Station. NOTE Details will be specified in the next revision of this part of ISO 24102.
FlowPolicy	Set flow policy for the currently available paths in local and neighbour ITS stations. NOTE Details will be specified in the next revision of this part of ISO 24102.
FlowFeedback	Request flow statistics from the network. NOTE Details will be specified in the next revision of this part of ISO 24102.
STAServDiscov	Discover an ITS station that can provide a service. NOTE Details will be specified in the next revision of this part of ISO 24102.

D.2 Description

D.2.1 Basics

This subclause defines the detailed structure of MN-COMMANDs.

D.2.2 SimNFcmd

Table D.2 — MN-COMMAND.request for MN-COMMAND SimNFcmd

ASN.1 type	Description
MN-Command.**simNFcmd**	Simulation of an NF-SAP service primitive issued at the ITS-S networking and transport layer to the ITS-S facilities layer. Requires activation of test mode.

D.2.3 FWTset

Table D.3 — MN-COMMAND.request for MN-COMMAND FWTset

ASN.1 type	Valid Range	Description
MN-Command.**fWTset**	CHOICE	Create a new entry in a forwarding table.

Table D.4 — MN-COMMAND.confirm for MN-COMMAND FWTset

ASN.1 type	Valid Range	Description
MN-CmdConfirm.**fWTsetConf**	CHOICE	Confirm creation of a new entry in a forwarding table.

D.2.4 FWTupdate

Table D.5 — MN-COMMAND.request for MN-COMMAND FWTupdate

ASN.1 type	Valid Range	Description
MN-Command.**fWTupdate**	CHOICE	Update an entry in a forwarding table.

D.2.5 FWTdelete

Table D.6 — MN-COMMAND.request for MN-COMMAND FWTdelete

ASN.1 type	Valid Range	Description
MN-Command.**fWTdelete**	CHOICE	Delete an entry in a forwarding table.

Annex E
(normative)

MN-REQUESTs

E.1 Overview

Table E.1 presents a summary of defined MN-COMMANDs. ASN.1 types are presented for .request/.confirm service primitives. A more detailed description is provided in E.2.

Table E.1 — MN-REQUESTs

REQUEST name	Description
SimNFreq	Request using service primitives of the NF-SAP towards the ITS-S management. Applicable only in test mode. ASN.1: `SimNFreq` / -
FWTsetNot	Notification of creation of an entry in a forwarding table. ASN.1: `FWTsetNot` / -
FWTupdateNot	Notification of an update of an entry in a forwarding table. ASN.1: `FWTupdateNot` / -
FWTdeleteNot	Notification of deletion of an entry in a forwarding table. ASN.1: `FWTdeleteNot` / -
VCIcreatePeerMAC	Request to create a VCI in a specific CI with a given relation to a peer station expressed by the MAC address of the peer station. ASN.1: `VCIcreatePeerMAC` / `VCIPeerMAC`
Its-ssiPeerNot	Notification of “ITS-SSI Data” from a peer station. ASN.1: `Its-ssiPeerNot` / NULL
STAGeoNot	Notification of geographic information of an ITS station. NOTE Details will be specified in the next revision of this part of ISO 241002.
STATopoNot	Notification of topological locator of an ITS station. NOTE Details will be specified in the next revision of this part of ISO 241002.
STAServNot	Notification of service of an ITS station. NOTE Details will be specified in the next revision of this part of ISO 241002.
PathNot	Notification of status of a path. NOTE Details will be specified in the next revision of this part of ISO 241002.
PathMetricNot	Notification of network metric of a path. NOTE Details will be specified in the next revision of this part of ISO 241002.
FlowStatistics	Provide network flow statistics. NOTE Details will be specified in the next revision of this part of ISO 241002.

E.2 Description

E.2.1 Basics

This subclause defines the detailed structure of MN-REQUESTs.

E.2.2 SimNFreq

Table E.2 — MN-REQUEST.request for MN-REQUEST SimNFreq

ASN.1 type	**Description**
MN-Request.**simNFreq**	Simulation of an NF-SAP service primitive issued at the ITS-S networking and transport layer to the ITS-S facilities layer. Requires activation of test mode.

This request shall not be acknowledged with MN-REQUEST.confirm.

E.2.3 FWTsetNot

Table E.3 — MN-REQUEST.request for MN-REQUEST FWTsetNot

ASN.1 type	**Description**
MN-Request.**fWTsetNot**	Notification of creation of an entry in a forwarding table.

This request shall not be acknowledged with MN-REQUEST.confirm.

E.2.4 FWTupdateNot

Table E.4 — MN-REQUEST.request for MN-REQUEST FWTupdateNot

ASN.1 type	**Description**
MN-Request.**fWTupdateNot**	Notification of an update of an entry in a forwarding table.

This request shall not be acknowledged with MN-REQUEST.confirm.

E.2.5 FWTdeleteNot

Table E.5 — MN-REQUEST.request for MN-REQUEST FWTdeleteNot

ASN.1 type	**Description**
MN-Request.**fWTdeleteNot**	Notification of deletion of an entry in a forwarding table.

This request shall not be acknowledged with MN-REQUEST.confirm.

E.2.6 VCIcreatePeerMAC

Table E.6 — MN-REQUEST.request for MN-REQUEST VCIcreatePeerMAC

ASN.1 type	**Description**
MN-Request.**vCIcreatePeerMAC**	Request to create a VCI in a specific CI with a given relation to a peer station expressed by the MAC address of the peer station.
VCIcreatePeerMAC.**reference**	Cyclic reference number. To be incremented with every next request.

Table E.6 *(continued)*

ASN.1 type	Description
VCIcreatePeerMAC.**linkID**	Link-ID of CI, in which the VCI shall be created.
VCIcreatePeerMAC.**mACaddress**	MAC address of a potential peer ITS station which likely is in an existing communication zone of the own station.

This request shall be acknowledged with MN-REQUEST.confirm as specified in Table E.7.

Table E.7 — MN-REQUEST.confirm for MN-REQUEST VCIcreatePeerMAC

ASN.1 type	Description
MN-ReqConfirm.**vCIPeerMAC**	Comfirms creation of the requested VCI.
VCIPeerMAC.**reference**	Same value as in related request.
VCIPeerMAC.**linkId**	Link-ID of new VCI specified in ISO 21218.

E.2.7 Its-ssiPeerNot

Table E.8 — MN-REQUEST.request for MN-REQUEST Its-ssiPeerNot

ASN.1 type	Description
MN-Request.**its-ssiPeerNot**	Notification of “ITS-SSI Data” from a peer station.
Its-ssiPeerNot.**sap**	SAP address indicating type of networking protocol.
Its-ssiPeerNot.**mACaddress**	MAC address of peer station which provided the “ITS-SSI Data”.
Its-ssiPeerNot.**linkId**	Link-ID related to peer station.
Its-ssiPeerNot.**its-ssiData**	ITS station state information specified in ISO 24102-1.

Annex F
(normative)

MI-COMMANDs

F.1 Overview

Table F.1 presents a summary of defined MN-COMMANDs. ASN.1 types are presented for .request service primitives. A more detailed description is provided in F.2. See also ISO 21218.

Table F.1 — MI-COMMANDs

COMMAND name	Description
SimINcmd	Command to simulate an access to the access layer via IN-SAP. Applicable only in test mode. ASN.1: `SimINcmd`
RegCmd	Command acknowledging the request to register the CI. ASN.1: `RegCmd`
CIstateChng	Change of CI status. ASN.1: `CIstateChng`
WakeUp	0: Stops transmission of wake-up signal. 1 to 255: Starts repetitive transmission of wake-up signal with maximum interval in milliseconds. ASN.1: `WakeUp`
RTScmd	Information on a Request To Send (RTS). Used for cross-CI prioritization. ASN.1: `RTScmd`
RTSackCmd	Acknowledgement of an RTSreq request. ASN.1: `RTSackCmd`
CONcmd	Request a CI of CI access class CIAC-2 to connect to the communication service as soon as possible. ASN.1: `CONcmd`
RIcmd	Provision of regulatory information. ASN.1: `RIcmd`
ManuCmd	Allows for manufacturer-specific access to the CI. Used e.g. for test and maintenance purposes. NOTE Details are outside the scope of this part of ISO 24102. ASN.1: `OCTET STRING`
VCIcmd	Command to request creation, reset, or deletion of a VCI. Setting of parameters different to the default values for a newly created VCI has to be done in subsequent MI-SET commands. ASN.1: `VciCmd`

Table F.1 *(continued)*

COMMAND name	Description
Monitor	Command to request monitoring of parameters. ASN.1: `Monitor`
UnitDataCmd	Command to request transmission of a management data packet. ASN.1: `UnitData`

F.2 Description

F.2.1 Basics

This subclause defines the detailed structure of MI-COMMANDs.

Some MI-COMMANDs are applicable only for selected media.

F.2.2 SimINcmd

MI-COMMAND “Simulate_IN-SAP_Cmd” shall be used by the ITS-S management to simulate an access to the access layer via IN-SAP.

ASN.1 type	Valid range	Description
MI-Command.**SimINcmd**		IN-SAP service primitive used by networking and transport layer.
SimINcmd.**simINcmd**	CHOICE	
simINcmd.**inDown**		IN-SAP service primitives (from ITS-S network and transport layer to ITS-S access layer).

F.2.3 RegCmd

MI-COMMAND “RegCmd” shall be used by the ITS-S management to acknowledge a registration request from a CI.

ASN.1 type	Valid range	Description
MI-Command.**regCmd**		Request to register the CI.
RegCmd.**scuId**	0 to 255	Value of ITS-SCU-ID assigned by ITS-S management.
RegCmd.**medID**	0 to 255	Value of MedID assigned by ITS-S management.

F.2.4 CIstateChng

MI-COMMAND “CIstateChng” shall be used by the ITS-S management to request a change of CI state.

ASN.1 type	Valid range	Description
MI-Command.**cIstate**	INTEGER	deregister activate reactivate connect disconnect suspend inactivate

F.2.5 WakeUp

MI-COMMAND “WakeUp” shall be used by the ITS-S management to enable and disable transmission of wake-up signals in a CI. Details depend on the access technology (medium).

ASN.1 type	Valid range	Description
MI-Command.**wakeUp**	0, 1 to 255	0: Stops transmission of wake-up signal. 1 to 255: Starts repetitive transmission of wake-up signal with maximum interval in milliseconds.

F.2.6 RTScmd

MI-COMMAND “RTScmd” shall be used by the ITS-S management to request prioritization over interfering CIs.

ASN.1 type	Valid range	Description
MI-Command.**rTScmd**	—	Information on a Request To Send (RTS) of another CI. Used for cross-CI prioritization. Shall be acknowledged by REQUEST “RTSackReq”.
RTScmd.**reqID**	1 to 255	LocalCIID of CI announcing the dummy request.
RTScmd.**priority**	0 to 255	User priority of dummy request.
RTScmd.**seqNo**	0 to 255	Sequential number to identify dummy request. Cyclic counter.
RTScmd.**status**	INTEGER	release request

F.2.7 RTSackCmd

MI-COMMAND “RTSackcmd” shall be used by the ITS-S management to acknowledge a prioritization request from a CI.

ASN.1 type	Valid range	Description
MI-Command.**rTSackCmd**	—	Acknowledgement of a REQUEST “RTSreq”.
RTSackCmd.**priority**	0 to 255	Equal to MinPrioCrossCI, see ISO 21218.
RTSackCmd.**seqNo**	0 to 255	Sequential number to identify dummy request. Cyclic counter.
RTSackCmd.**status**	INTEGER	ignored granted

F.2.8 CONcmd

MI-COMMAND "CONcmd" shall be used by the CALM management to request that a CI of CI access class CIAC-2 or CIAC-3 connects/disconnects to the communication service or deletes access information.

ASN.1 type	Valid range	Description
MI-Command.**cONcmd**	INTEGER	deleteAC: Delete access information I-parameters "SIMpin" and "ProviderInfo". This shall automatically disconnect from the service, if applicable. connect: Connect asap with given access information. If this information is not available, confirm command with error code 7. disconnect: Disconnect immediately without deleting access information.

F.2.9 RIcmd

MI-COMMAND "RIcmd" shall be used by the ITS-S management to forward a management data packet to a CI containing regulatory information as received via another CI.

ASN.1 type	Valid range	Description
MI-Command.**rIcmd**		
RIcmd.**linkID**		Link-ID of CI which shall receive regulatory information.
RIcmd.**ri**	OCTET STRING	Regulatory Information (RI) in medium-specific format.

F.2.10 Manufacturer

MI-COMMAND "ManuCmd" shall be used by the ITS-S management to allow a manufacturer to have private access to their CI.

ASN.1 type	Valid range	Description
MI-Command.**manuCmd**	any octet string	Manufacturer-specific octet string.

F.2.11 VCIcmd

MI-COMMAND "VCIcmd" shall be used by the ITS-S management to request creation, deletion, or reset of a VCI.

ASN.1 type	Valid range	Description
MI-Command.**VCIcmd**		
VCIcmd.**linkID**		Link-ID.
VCIcmd.**alive**	BOOLEAN	FALSE: Delete VCI, if alive. TRUE: Create VCI, if not existent, or reset it. For creation. use vciRef as SerialNumber of the LINK-ID of the new VCI .

F.2.12 Monitor

MI-COMMAND "Monitor" shall be used by the ITS-S management to request automatic notification of change of parameter values.

ASN.1 type	Valid range	Description
MI-Command.**monitor**	SEQUENCE	Sequence of parameters of which change of value shall be notified.
Monitor.**I-ParamNo**	CHOICE tag	Reference tag number of parameter to be monitored.
Monitor.**active**	0, 255	0: stop monitoring 255: start monitoring

F.2.13 UnitDataCmd

MI-COMMAND "UnitDataCmd" shall be used by the ITS-S management to request transmission of a management data packet.

ASN.1 type	Valid range	Description
MI-Command.**unitDataCmd**		Transmission request of a management data packet.
UnitDataCmd.**sourceAddr**	Link-ID	Same as source_address in DL_UNITDATA.request.
UnitDataCmd.**destAddr**	Link-ID	Same as destination_address in DL_UNITDATA.request.
UnitDataCmd.**data**		Same as data in DL_UNITDATA.request.
UnitDataCmd.**priority**		Same as priority in DL_UNITDATA.request.
UnitDataCmd.**parameter**		Parameters to be set prior to transmission/status information of received packet. Details depend on the access technology (medium).

Annex G
(normative)

MI-REQUESTs

G.1 Overview

Table G.1 presents a summary of defined MN-COMMANDs. ASN.1 types are presented for .request service primitives. A more detailed description is provided in G.2. See also ISO 21218.

Table G.1 — MI-REQUESTs

REQUEST name	**Description**
SimINreq	Request using service primitives of the IN-SAP towards the ITS-S management. Applicable only in test mode. ASN.1: `SimINreq`
RegReq	Request to register the CI. ASN.1: `RegReq`
PrioReg	Cross-CI Prioritization Registration. ASN.1: `PrioReg`
RTSreq	Dummy Request To Send (RTS). Used for cross-CI prioritization. ASN.1: `RTSackReq`
RTSackReq	Acknowledgement of COMMAND “RTScmd”. Used for cross-CI prioritization. ASN.1: `RTSackReq`
RIreq	Request to get regulatory information via another CI. ASN.1: `RIreq`
ManuReq	Optional reply to a COMMAND “ManuCmd”. ASN.1: `OCTET STRING`
Events	Notification of an event. ASN.1: `Events21218`
PosUpdateReq	Requests to receive position updates with update interval as indicated in milliseconds/cancels the request. ASN.1: `PosUpdateReq`
UnitDataReq	Reception notification of a management data packet. ASN.1: `UnitData`

G.2 Description

G.2.1 Basics

This subclause defines the detailed structure of MI-REQUESTs.

Some REQUESTs are applicable only for selected media.

G.2.2 SimINreq

MI-REQUEST “SimINreq” shall be used by

ASN.1 type	Valid range	Description
MI-Request.**simINreq**		
SimINreq.**simINreq**		Service primitive of the IN-SAP used by access layer.
simINreq.**inUp**		IN-SAP service primitives (from ITS-S access layer to ITS-S network and transport layer).

G.2.3 RegReq

MI-REQUEST “RegReq” shall be used by the MAE to register a CI.

ASN.1 type	Valid range	Description
MI-Request.**regReq**		Request to register the CI.
RegReq.**medType**	Same as I-Parameter “MedType”	Access technology (medium) of the CI sending the command.

G.2.4 PrioReg

MI-REQUEST “PrioritizationRequest” shall be used by the MAE to register at the ITS-S management for the cross-CI prioritization procedure.

ASN.1 type	Valid range	Description
MI-Request.**prioReg**		Managing of “Cross-CI prioritization” in the interfering CI.
PrioReg.**interferers**	Sequence of I-Parameter “MedType	MedType of potential interferers. Known *a priori* to CI.
PrioReg.**timeout**	1 to 255	Timeout in milliseconds. Set to a default value by CI.

G.2.5 RTSreq

MI-REQUEST “RTSreq” shall be used by the MAE to request and release cross-CI prioritization.

ASN.1 type	Valid range	Description
MI-Request.**rTSreq**	—	Dummy Request To Send (RTS). Used for cross-CI prioritization.
RTSreq.**priority**	0 to 255	User priority of dummy request.
RTSreq.**seqNo**	0 to 255	Sequential number to identify dummy request. Cyclic counter.
RTSreq.**status**	INTEGER	release request

G.2.6 RTSackReq

MI-REQUEST “RTSackReq” shall be used by the MAE to acknowledge a cross-CI prioritization request.

ASN.1 type	Valid range	Description
MI-Request.**rTSackReq**	—	Dummy Request To Send (RTS). Used for cross-CI prioritization.
RTSackReq.**reqID**	SEQUENCE	LocalCIID of CI requesting prioritization.

ASN.1 type	Valid range	Description
RTSackReq.**seqNo**	0 to 255	Sequential number to identify dummy request. Cyclic counter.
RTSackReq.**status**	INTEGER	ignored granted

G.2.7 RIreq

MI-REQUEST “RIreq” shall be used by the MAE to request retrieval of regulatory information via another CI.

ASN.1 type	Valid range	Description
MI-Request.**riReq**		
RiReq.**medType**	MedType	Access technology (medium) to be used to retrieve regulatory information.
RiReq.**riAccess**	Any octet string	Access information needed.

G.2.8 ManuReq

MI-REQUEST “ManuReq” shall be used by the MAE to reply to a manufacturer-specific access to the CI.

ASN.1 type	Valid range	Description
MI-Request.**manuReq**	Any octet string	Manufacturer specific octet string. Optional reply to a COMMAND “ManuCmd”.

G.2.9 Events

MI-REQUEST “Events” shall be used by the MAE to notify events.

ASN.1 type	Valid range	Description
MI-Request.**events**	CHOICE	Event as specified in Table G.2

Table G.2 — Events

Event	Event description	Request.Value
E21218-0	A transmission request was rejected due to a user priority that was below the minimum required priority as defined by I-Parameter “MinimumUser-Priority”.	I-Parameter “MinimumUserPriority” Link-ID of VCI
E21218-1	A transmission queue is filled above the threshold defined by I-Parameter “QueueAlarmThreshold”.	Priority of the queue
E21218-2	A transmission queue is full.	Priority of the queue
E21218-3	A VCI was created.	Link-ID of VCI
E21218-4	A VCI was deleted.	Link-ID of VCI
E21218-5	An I-Parameter subject to notification has changed its value.	I-Param
E21218-6	A transmission queue is emptied below the threshold defined by I-Parameter “QueueLowThreshold”.	Priority of the queue
E21218-7	A VCI was reset.	Link-ID of VCI

G.2.10 PosUpdateReq

MI-REQUEST “PosUpdateReq” shall be used by the MAE to manage reception of updates of the actual position of the station.

ASN.1 type	Valid range	Description
MI-Request.**posUpdateReq**	0 to 65 535	0: Stop updates. >0: Updates with interval given in milliseconds.

A value zero provided in the MI-COMMAND shall stop delivery of updates by the ITS-S management.

A value larger than zero in the MI-COMMAND shall start delivery of updates by the ITS-S management with an update interval in milliseconds as indicated by the value provided.

The updates shall be written in I-Parameter “KinematicVector” by means of the SET service.

G.2.11 UnitDataReq

MI-REQUEST “UnitDataReq” shall be used by the MAE to notify reception of management data packets.

ASN.1 type	Valid range	Description
MI-Request.**unitDataReq**		Reception notification of a management data packet.
UnitDataReq.**sourceAddr**	Link-ID	Same as source_address in DL_UNITDATA.indication.
UnitDataReq.**destAddr**	Link-ID	Same as destination_address in DL_UNITDATA.indication.
UnitDataReq.**data**		Same as data in DL_UNITDATA.indication.
UnitDataReq.**priority**		Same as priority in DL_UNITDATA.indication.
UnitDataReq.**parameter**		Parameters to be set prior to transmission/status information of received packet. Details depend on the access technology (medium).

Annex H
(normative)

SF-COMMANDs

NOTE This normative annex will be provided in a future version of this part of ISO 24102.

Annex I
(normative)

SF-REQUESTs

NOTE This normative annex will be provided in a future version of this part of ISO 24102.

Annex J
(normative)

SN-COMMANDs

NOTE This normative annex will be provided in a future version of this part of ISO 24102.

Annex K
(normative)

SN-REQUESTs

NOTE This normative annex will be provided in a future version of this part of ISO 24102

Annex L
(normative)

SI-COMMANDs

NOTE This normative annex will be provided in a future version of this part of ISO 24102.

Annex M
(normative)

SI-REQUESTs

NOTE This normative annex will be provided in a future version of this part of ISO 24102.

Annex N
(normative)

MS-COMMANDs

NOTE This normative annex will be provided in a future version of this part of ISO 24102.

Annex O
(normative)

MS-REQUESTs

NOTE This normative annex will be provided in a future version of this part of ISO 24102.

Annex P
(normative)

Error/return codes

Error/return codes shall be as follows in Table P.1.

Table P.1 — Error/return codes

Valid range	Description
INTEGER	0: SUCCESS 1: UNSPECIFIED FAILURE 2: INVALID PARAMETER NUMBER 3: INVALID PARAMETER VALUE 4: RI VIOLATION 5: INVALID COMMAND/REQUEST NUMBER 6: INVALID COMMAND/REQUEST VALUE 7: ACCESS VIOLATION 8: INVALID COMMAND/REQUEST TYPE, 9: SEQUENCE ERROR 10: VALUE NOT AVAILABLE 255: UNSPECIFIED HARDWARE FAILURE

Bibliography

[1] ISO 24102-2, *Intelligent transport systems — Communications access for land mobiles (CALM) — Remote ITS station management*

[2] ISO 17423, *Intelligent transport systems — Cooperative systems — Application requirements for selection of communication profiles*

ICS 03.220.01;35.240.60

Price based on 65 pages

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

All the data, software and documentation set out in all British Standards and other BSI publications are the property of and copyrighted by BSI, or some person or entity that owns copyright in the information used (such as the international standardization bodies) and has formally licensed such information to BSI for commercial publication and use. Except as permitted under the Copyright, Designs and Patents Act 1988 no extract may be reproduced, stored in a retrieval system or transmitted in any form or by any means – electronic, photocopying, recording or otherwise – without prior written permission from BSI. Details and advice can be obtained from the Copyright & Licensing Department.

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

bsi.

...making excellence a habit.™